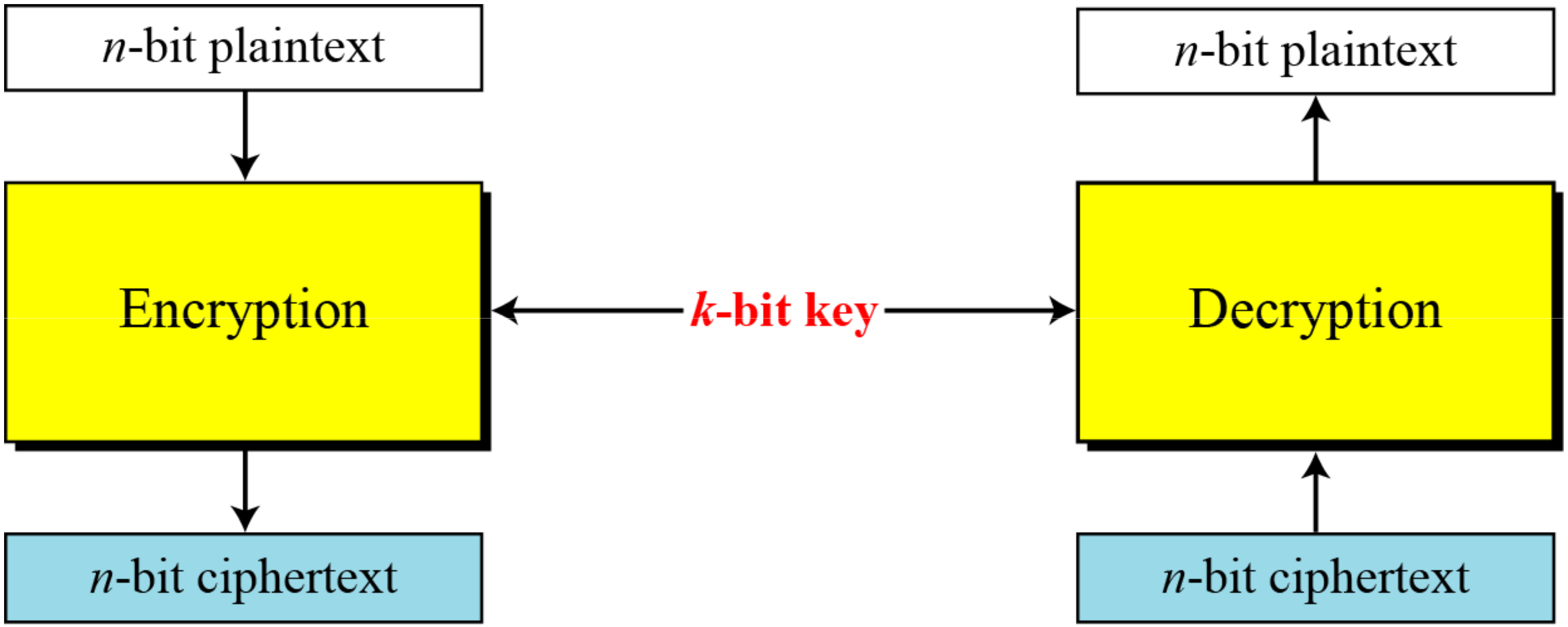
Introduction to Modern Symmetric key ciphers

Modern Block ciphers

A symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext. The encryption or decryption algorithm uses a k-bit key.

Modern Block ciphers…



Modern Block ciphers…

- Example
  - How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

Modern Block ciphers...

- Example
  - How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?
- Solution
  - Encoding 100 characters using 8-bit ASCII results in an 800-bit message. The plaintext must be divisible by 64. If | M | and |Pad| are the length of the message and the length of the padding,

$$|M| + |Pad| = 0 \bmod 64 \quad \rightarrow \quad |Pad| = -800 \bmod 64 \quad \rightarrow \quad 32 \bmod 64$$

Substitution or Transposition

- A modern block cipher can be designed to act as a substitution cipher or a transposition cipher.
- Which one do you think is better? Why?

Substitution or Transposition…

- Let us take an example
  - Suppose that we have a block cipher where n = 64. If there there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?

a. The cipher is designed as a substitution cipher.

b. The cipher is designed as a transposition cipher.

Substitution or Transposition…

- Solution
  - In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.
    - Would take hundreds of years if Eve would try 1 billion blocks per second !!!
  - In the second case, Eve knows that there are exactly 10 1's in the plaintext. Eve can launch an exhaustive-search attack using only those 64-bit blocks that have exactly 10 1's.
    - Would need to try (64!)/[(10!)(54!)] possibilities
    - Can be successful in less than 3 minutes

Substitution or Transposition

To be resistant to exhaustive-search attack,
a modern block cipher needs to be designed as a
substitution cipher.

Block Ciphers as Permutation Groups

- Is a modern block cipher a group?
  - Full-Size Key Transposition Block Ciphers
    - In a full-size key transposition cipher we need to have n! possible keys, so the key should have $\lceil \log_2(n!) \rceil$ bits.

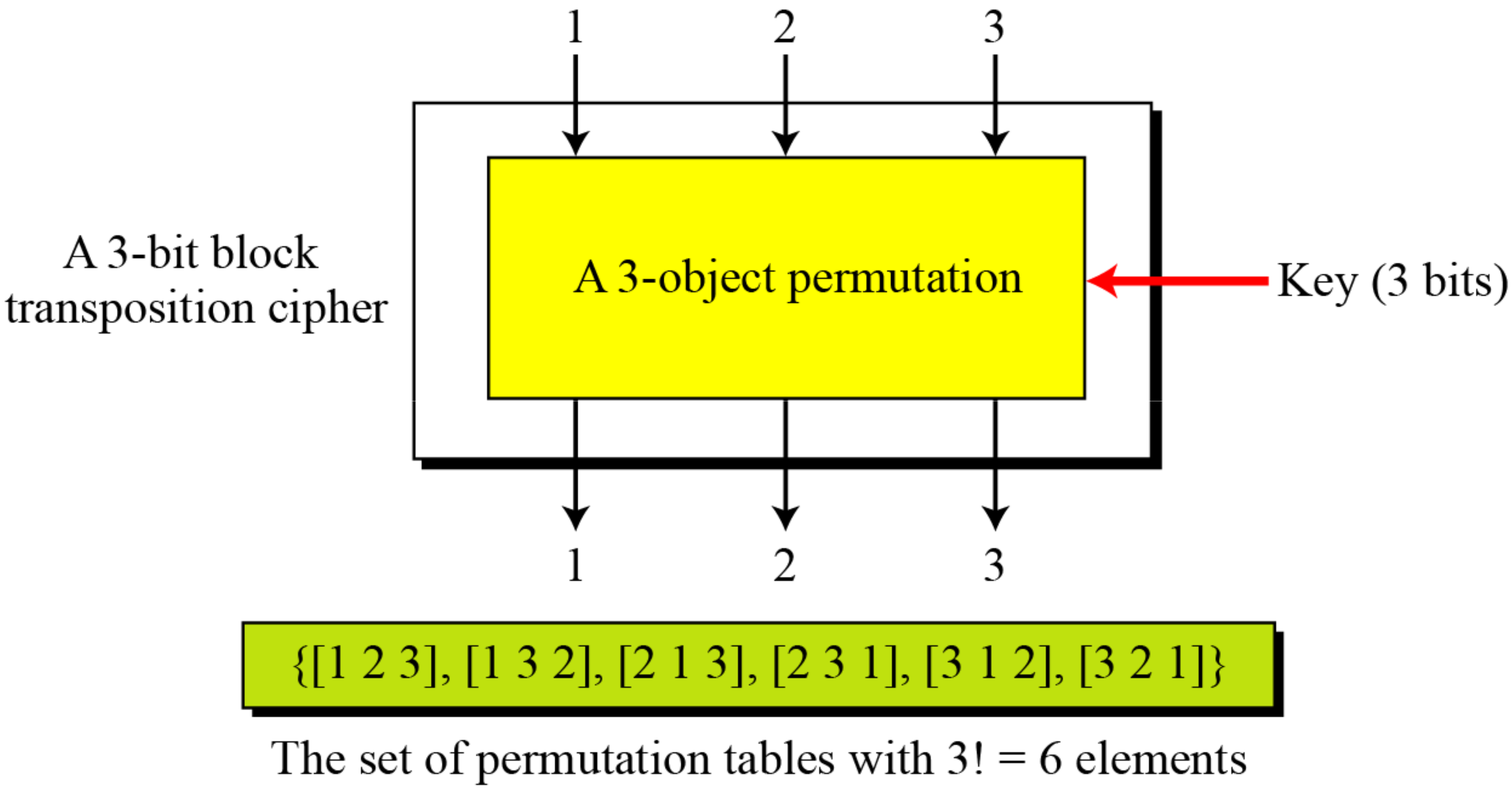
Example

Show the model and the set of permutation tables for a 3-bit block transposition cipher where the block size is 3 bits.

Solution

The set of permutation tables has 3! = 6 elements

Block Ciphers as Permutation Groups...



Block Ciphers as Permutation Groups...

- Full-Size Key substitution Block Ciphers
  - A full-size key substitution cipher does not transpose bits; it substitutes bits.
  - We can model the substitution cipher as a permutation if we can decode the input and encode the output.

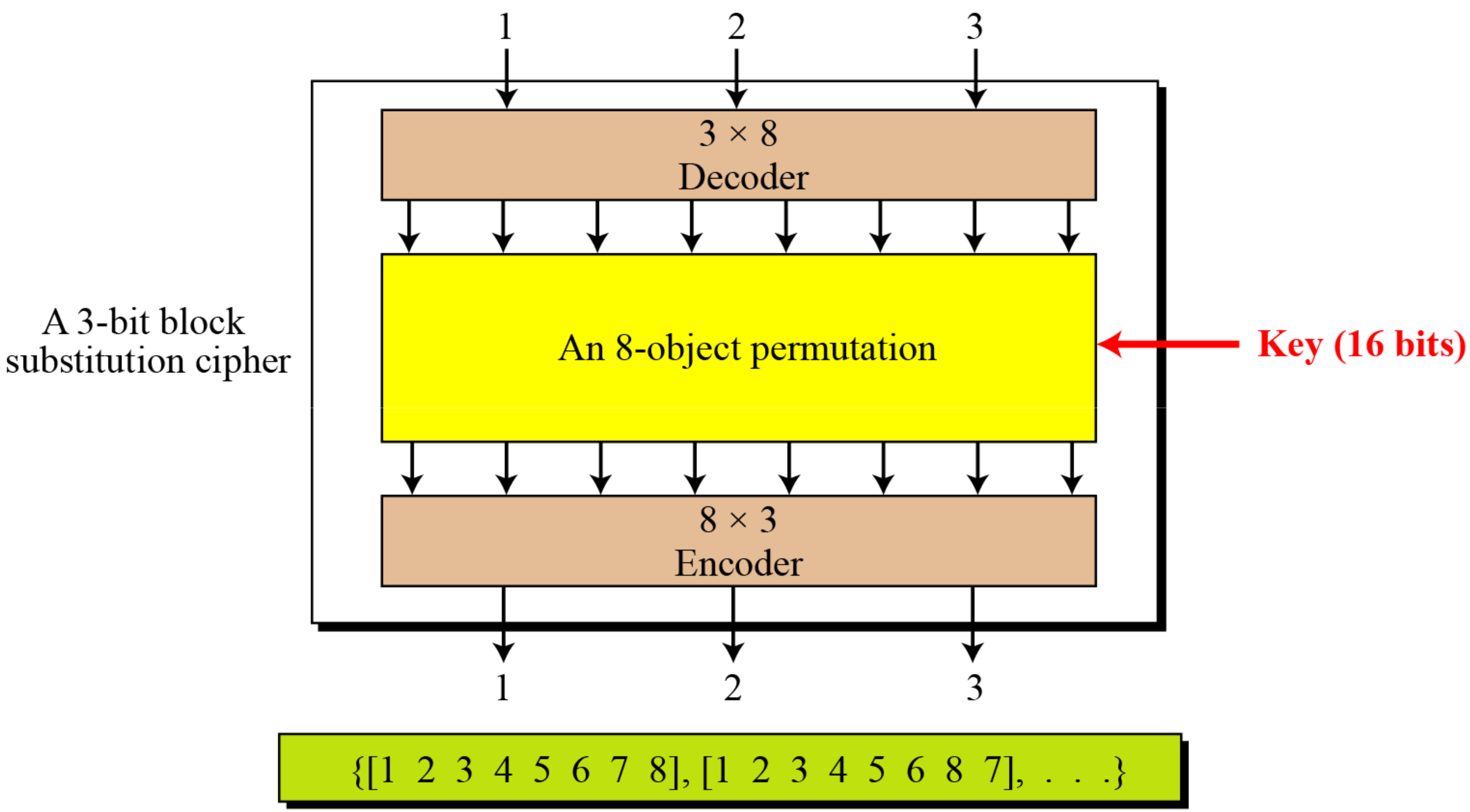
Example:

Show the model and the set of permutation tables for a 3-bit block substitution cipher.

Solution

Leads to a much longer key of $\lceil \log_2 (40320) \rceil = 16$ bits

Block Ciphers as Permutation Groups...



The set of permutation tables with 8! = 40,320 elements

Block Ciphers as Permutation Groups...

- A full-size key n-bit transposition cipher or a substitution block cipher can be modeled as a permutation, but their key sizes are different:
  - Transposition: the key is ???
  - Substitution: the key is ???

Block Ciphers as Permutation Groups...

- A full-size key n-bit transposition cipher or a substitution block cipher can be modeled as a permutation, but their key sizes are different:
  - Transposition: the key is $\log_2(n!)$
  - Substitution: the key is $\log_2(2^n!)$

Block Ciphers as Permutation Groups...

- It is useless to have more than one stage of full size key ciphers, because the effect is the same as having a single stage
- Can you justify this ???

Block Ciphers as Permutation Groups...

- Partial size key ciphers
  - Actual ciphers can not use full-size keys because the size of the key becomes large
  - E.g. DES is a common substitution cipher with 64-bit block
    - If the designers of DES would have used full-size key cipher, what will be the size of a key?

Block Ciphers as Permutation Groups...

- Analysis of full-size key ciphers
  - Actual ciphers can not use full-size keys because the size of the key becomes large
  - E.g. DES is a common substitution cipher with 64-bit block
    - If the designers of DES would have used full-size key cipher, what will be the size of a key?
      - The answer is $\log_2(2^{64}!) \approx 2^{70}$ bits !!!!
      - DES uses only 56-bit keys ....

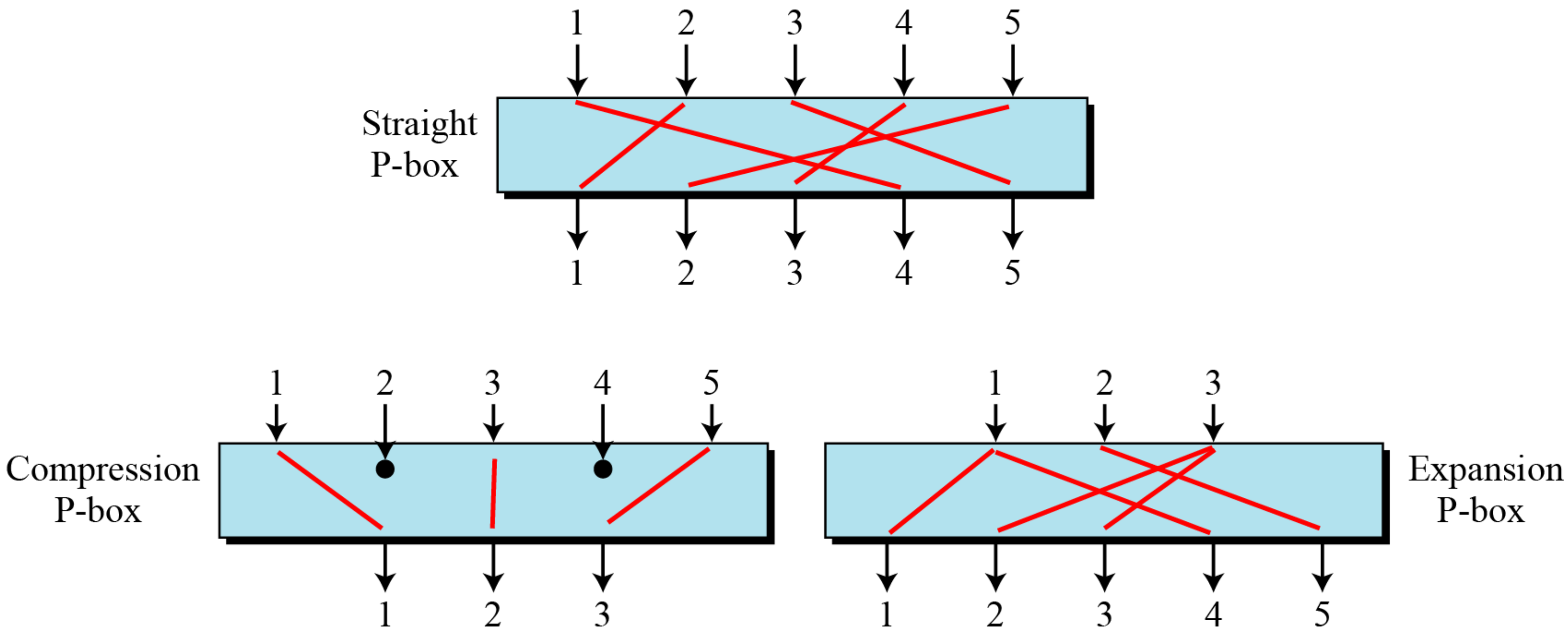
Block Ciphers as Permutation Groups...

- Partial-size key ciphers
  - Is a group under the composition operation if it is a subgroup of the corresponding full-size cipher
  - Can a multistage version of a partial-size key cipher be made to achieve more security?

Components of a Modern Block Cipher

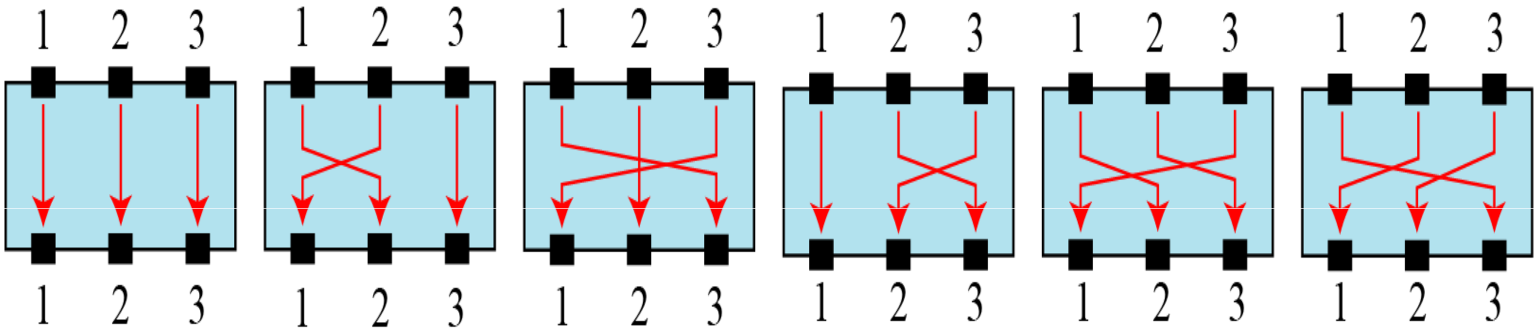
- Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.
- P-Boxes
  - A P-box (permutation box) parallels the traditional transposition cipher for characters.
  - It transposes bits.

Components of a Modern Block Cipher...



Components of a Modern Block Cipher...

- Possible mappings of a P-Box



Components of a Modern Block Cipher...

- Straight P-Box

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

Components of a Modern Block Cipher...

- Example
  - Design an 8×8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.

Components of a Modern Block Cipher...

- Example
  - Design an 8 × 8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.
- Solution
  - We need a straight P-box with the table [4 1 2 3 6 7 8 5].

Components of a Modern Block Cipher...

- Compression P-Boxes
  - A compression P-box is a P-box with n inputs and m outputs where m < n.
  - Example of a 32 x 24 permutation table

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

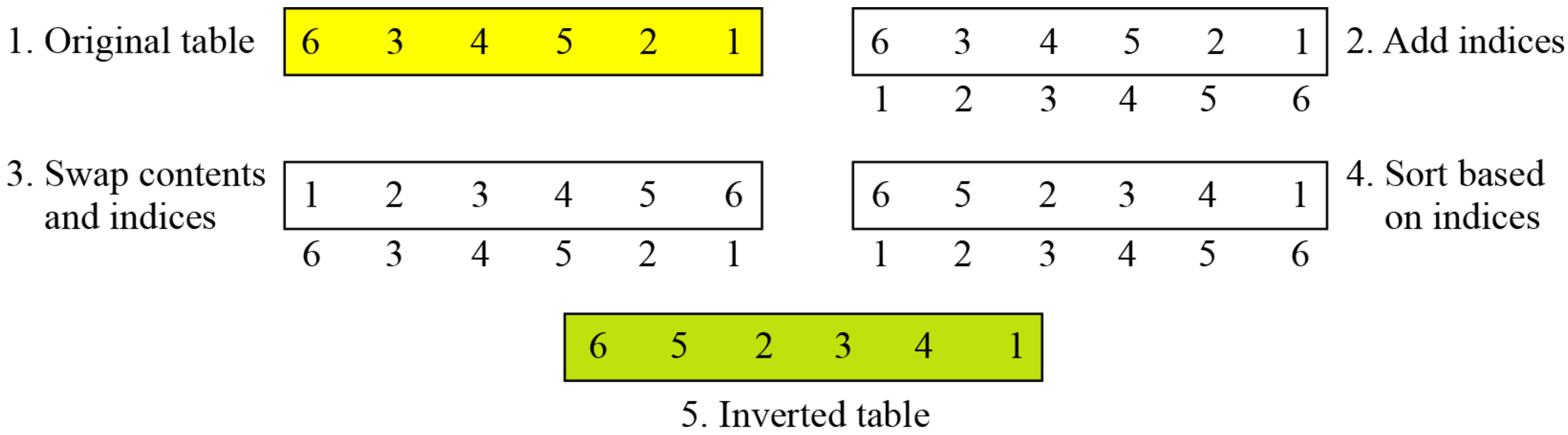
Components of a Modern Block Cipher...

- Expansion P-Boxes
  - An expansion P-box is a P-box with n inputs and m outputs where m > n.
  - Example of a 12 X 16 P-Box

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12

Components of a Modern Block Cipher...

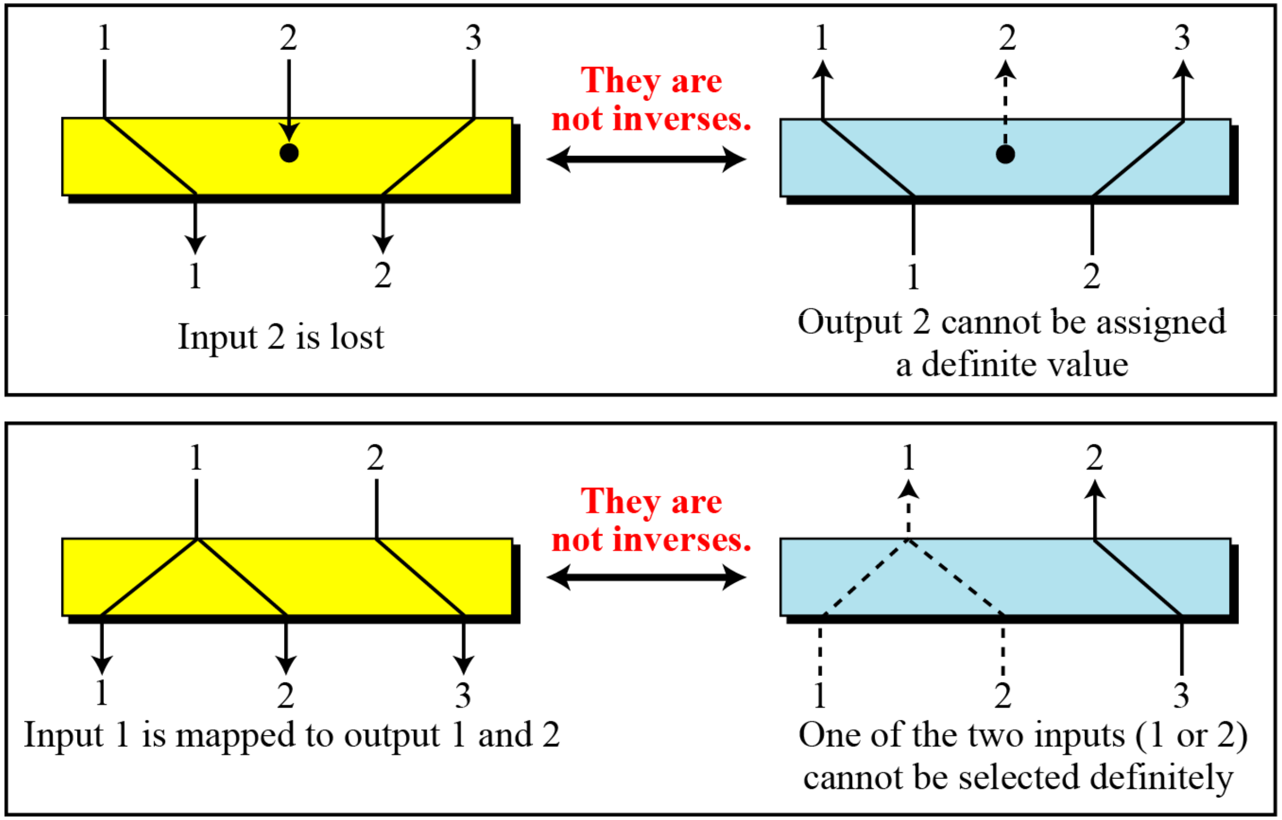
- P-Box invertibility
  - What can you say about this???
  - A straight P-box is invertible, but compression and expansion P-boxes are not.
- Inverting a permutation table represented as a one-dimensional table.



Components of a Modern Block Cipher...

- Compression and expansion P-boxes are non-invertible

Compression P-box



Expansion P-box

Components of a Modern Block Cipher...

- S-Box
  - An S-box (substitution box) can be thought of as a miniature substitution cipher.
  - An S-box is an m × n substitution unit, where m and n are not necessarily the same.
  - Example
    - In an S-box with three inputs and two outputs, we have,

$$y_1 = x_1 \oplus x_2 \oplus x_3 \qquad y_2 = x_1$$

Components of a Modern Block Cipher...

- S-Box
  - The S-box is linear because $a_{1,1} = a_{1,2} = a_{1,3} = a_{2,1} = 1$ and
    $a_{2,2} = a_{2,3} = 0$. The relationship can be represented by matrices, as shown below:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Components of a Modern Block Cipher...

- S-Box
  - In an S-box with three inputs and two outputs, we have

$$y_1 = (x_1)^3 + x_2 \qquad y_2 = (x_1)^2 + x_1x_2 + x_3$$

  - where multiplication and addition is in GF(2). The S-box is nonlinear because there is no linear relationship between the inputs and the outputs.

Components of a Modern Block Cipher...

- Example
  - The following table defines the input/output relationship for an S-box of size 3 × 2. The leftmost bit of the input defines the row; the two rightmost bits of the input define the column. The two output bits are values on the cross section of the selected row and column.

Leftmost bit

Rightmost bits

	00	01	10	11
0	00	10	01	11
1	10	00	11	01

Output bits

  - Based on the table, an input of 010 yields the output 01. An input of 101 yields the output of 00.

Components of a Modern Block Cipher...

- S-Box Invertibility
  - An S-box may or may not be invertible.
  - In an invertible S-box, the number of input bits should be the same as the number of output bits.

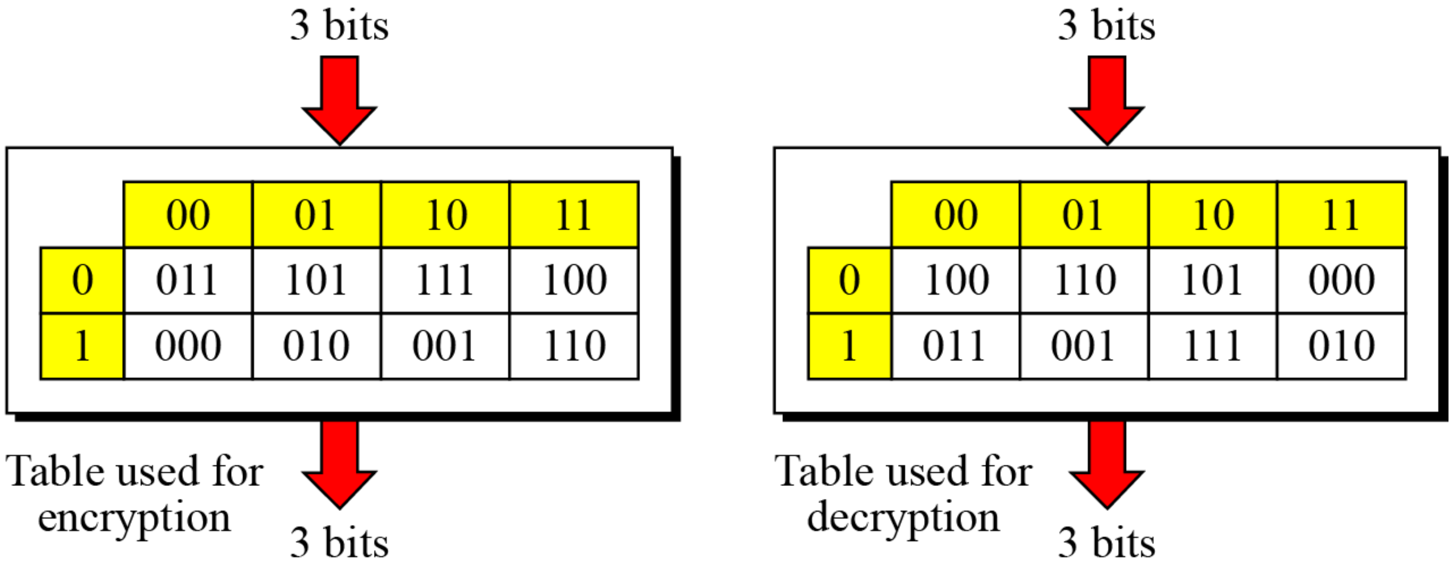


Table used for encryption

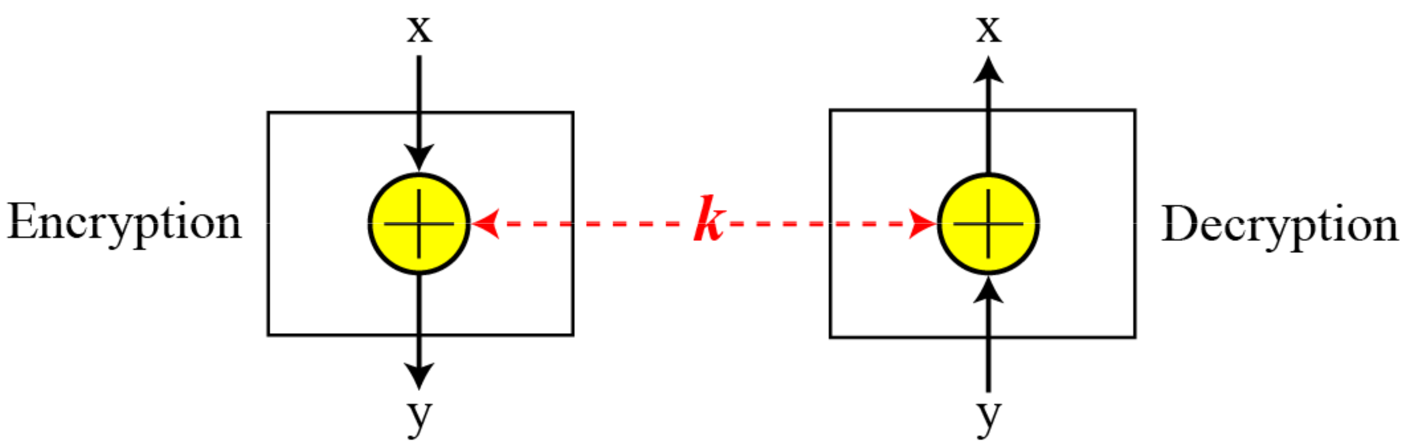
	00	01	10	11
0	011	101	111	100
1	000	010	001	110

Table used for decryption

	00	01	10	11
0	100	110	101	000
1	011	001	111	010

Components of a Modern Block Cipher...

- Exclusive-Or



Components of a Modern Block Cipher...

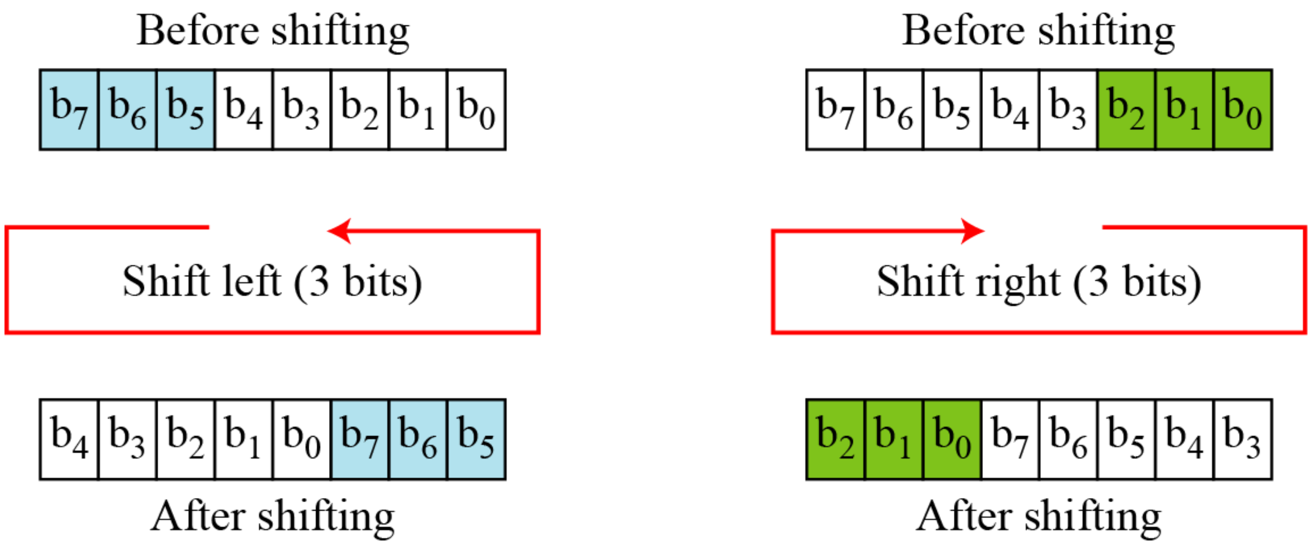
- Exclusive-Or...
  - An important component in most block ciphers is the exclusive-or operation.
  - Addition and subtraction operations in the $GF(2^n)$ field are performed by a single operation called the exclusive-or (XOR).
  - The five properties of the exclusive-or operation in the $GF(2^n)$ field makes this operation a very interesting component for use in a block cipher: closure, associativity, commutativity, existence of identity, and existence of inverse.

Components of a Modern Block Cipher...

- Exclusive-Or...
  - The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).
    - For example, a keyless P-box or a keyless S-box can be made invertible because they have one input and one output.
    - An exclusive operation is a binary operation. The inverse of an exclusive-or operation can make sense only if one of the inputs is fixed (is the same in encryption and decryption).
    - For example, if one of the inputs is the key, which normally is the same in encryption and decryption, then an exclusive-or operation is self-invertible

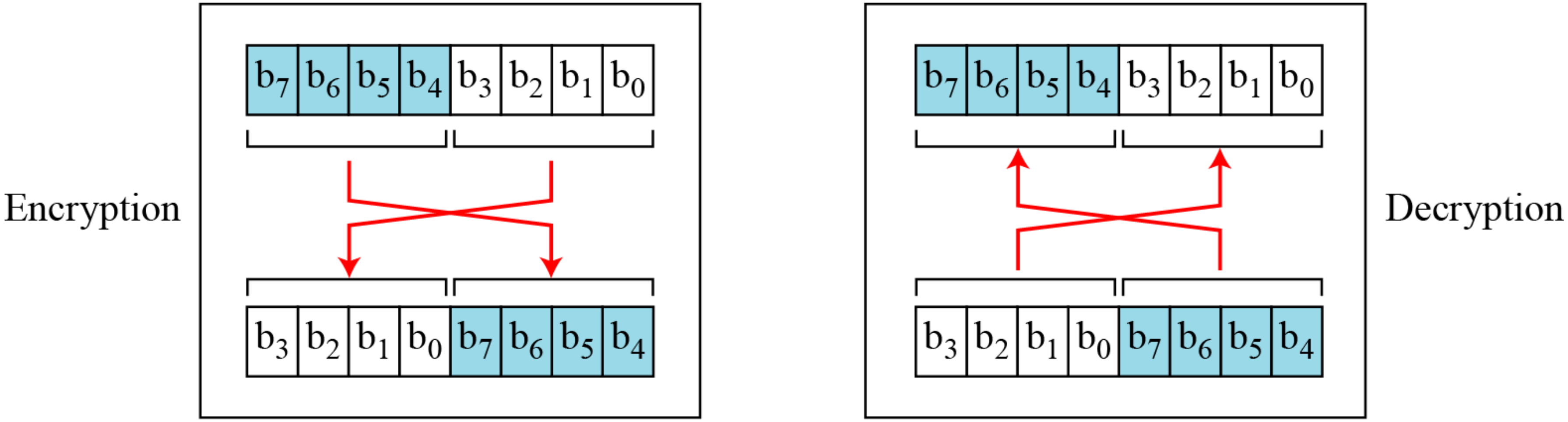
Components of a Modern Block Cipher...

- Circular Shift operation



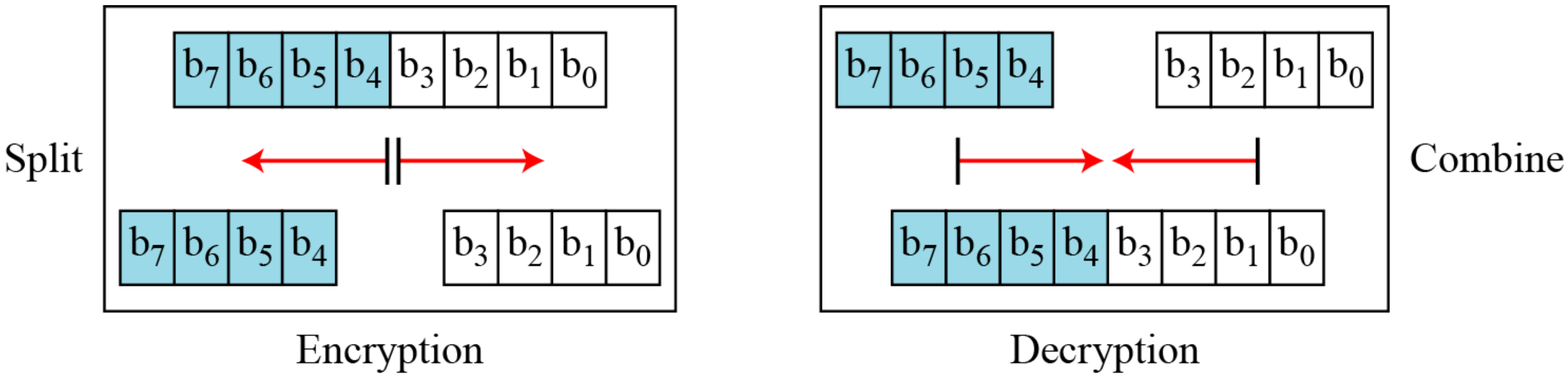
Components of a Modern Block Cipher...

- Swap
  - The swap operation is a special case of the circular shift operation where k = n/2.



Components of a Modern Block Cipher...

- Split and Combine
  - Two other operations found in some block ciphers are split and combine.



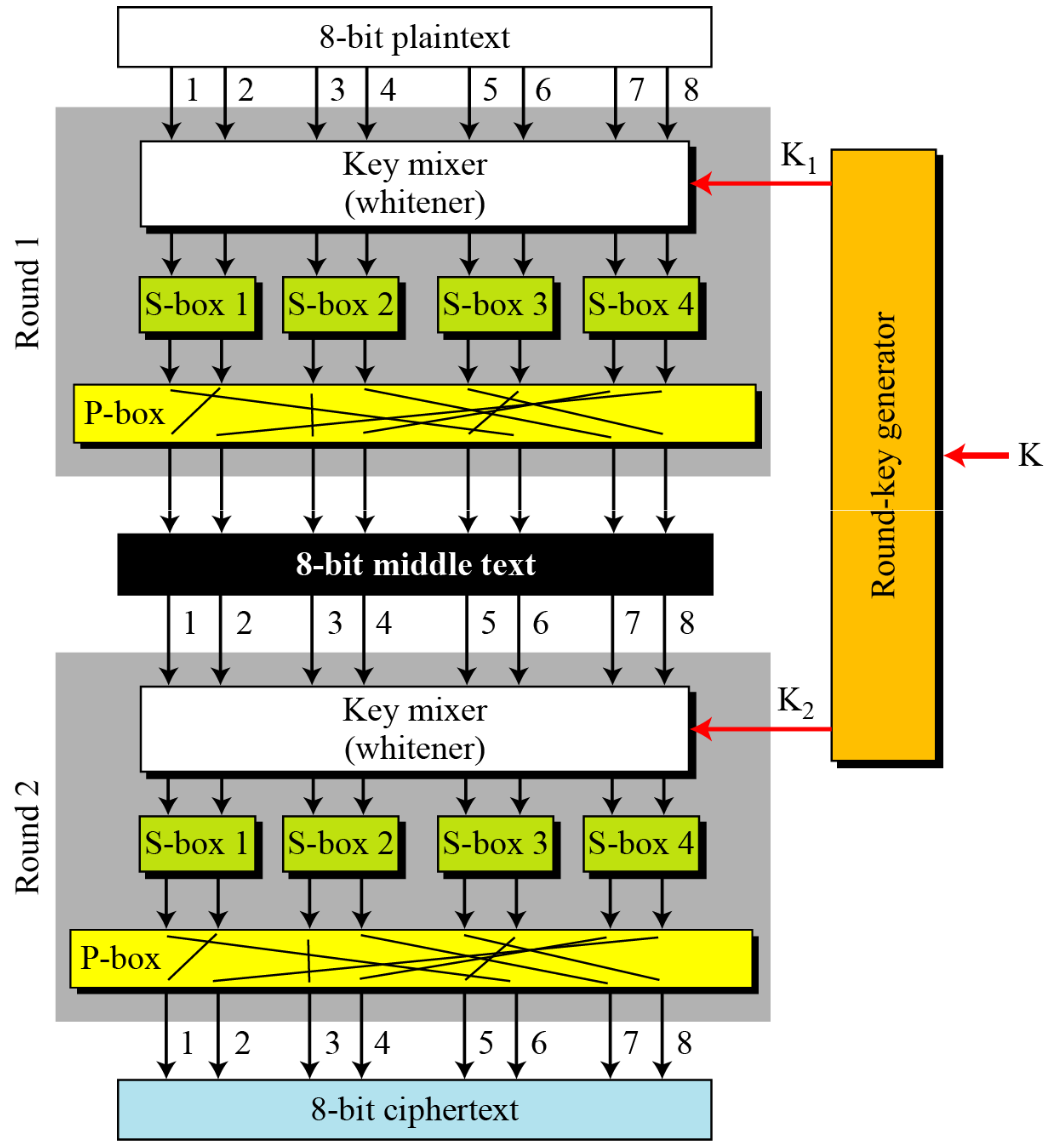
Product cipher

- Shannon introduced the concept of a product cipher.
- A product cipher is a complex cipher combining substitution, permutation, and other components

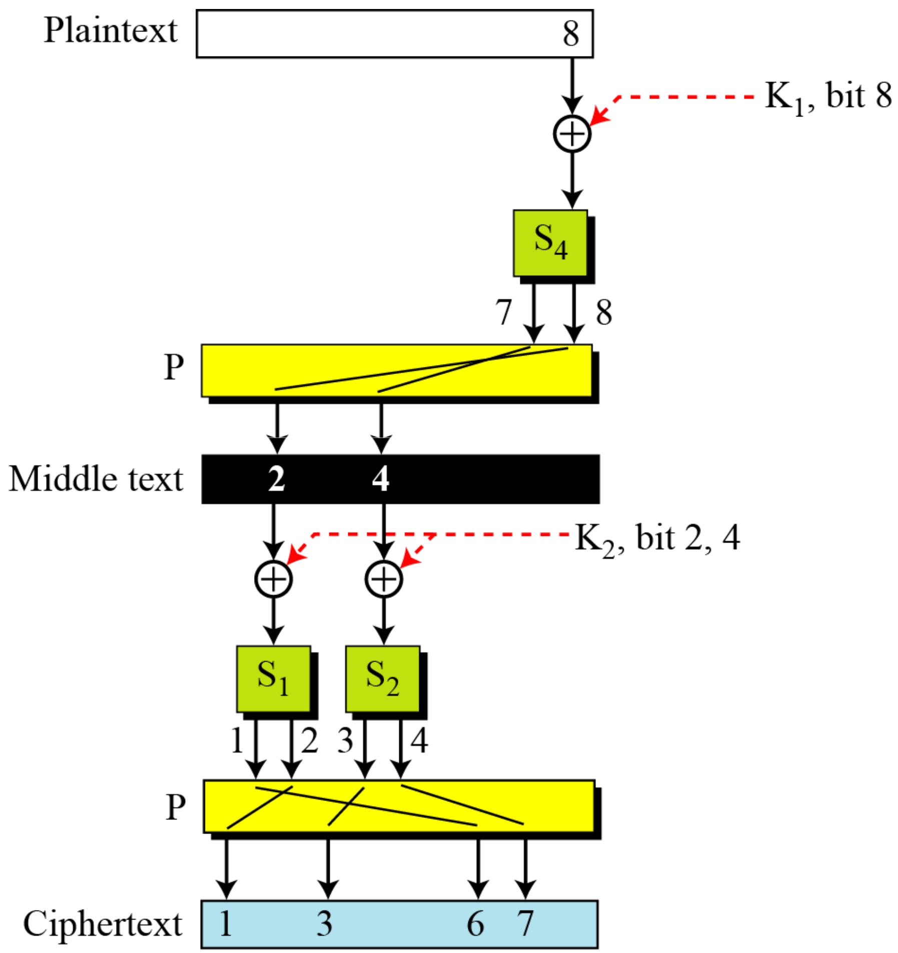
Product cipher...

- Diffusion
  - The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.
- Confusion
  - The idea of confusion is to hide the relationship between the ciphertext and the key.
- Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.

Product cipher...



Product cipher...



Two classes of product ciphers

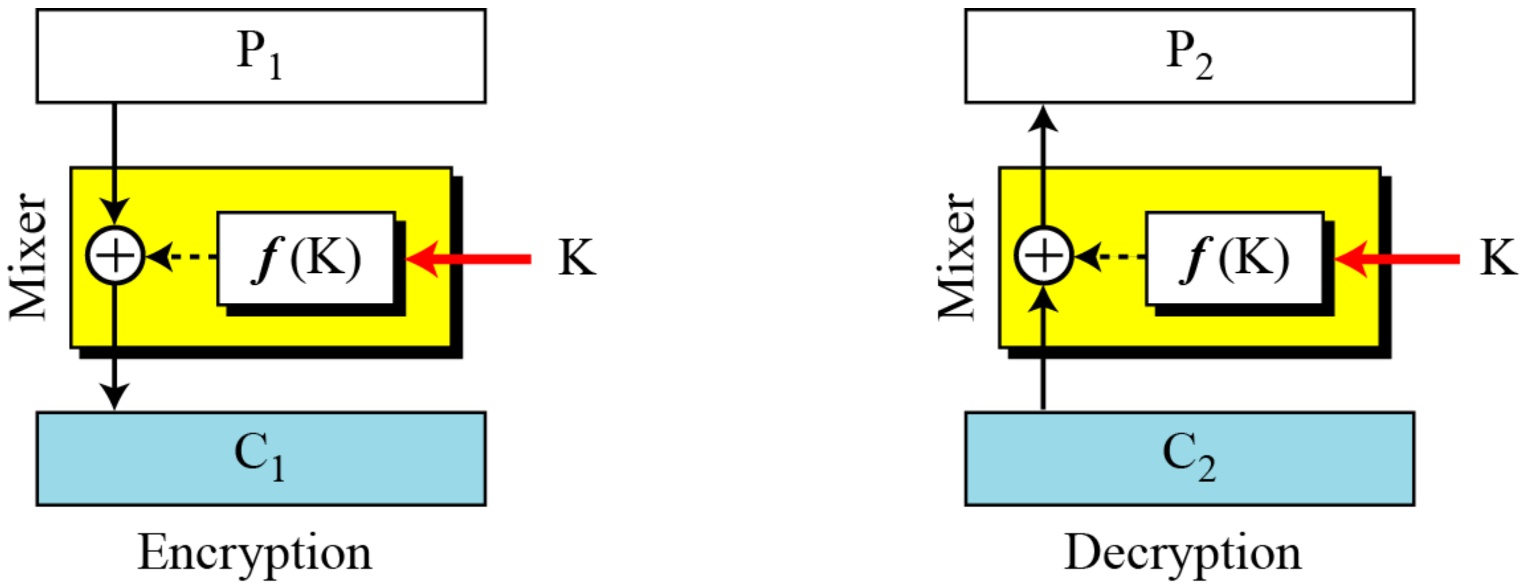
- Modern block ciphers are all product ciphers, but they are divided into two classes.
  - Feistel ciphers
  - Non-Feistel ciphers

Two classes of product ciphers...

- Feistel Ciphers
  - Feistel designed a very intelligent and interesting cipher that has been used for decades.
  - A Feistel cipher can have three types of components: self-invertible, invertible, and noninvertible.

Two classes of product ciphers...

- The first thought in Feistel Cipher design



Two classes of product ciphers...

- Example
  - The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

Two classes of product ciphers...

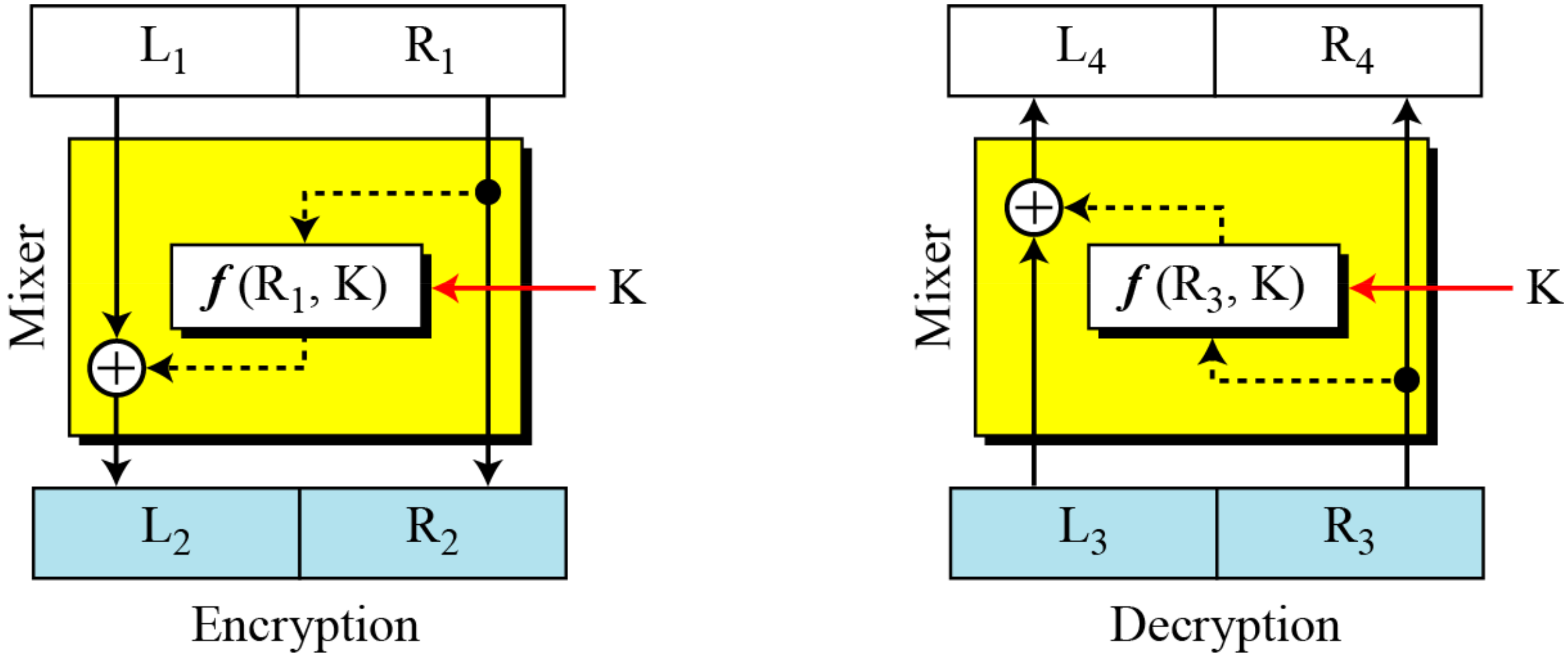
- Solution
  - The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

Encryption: $C = P \oplus f\,(K) = 0111 \oplus 1001 = 1110$

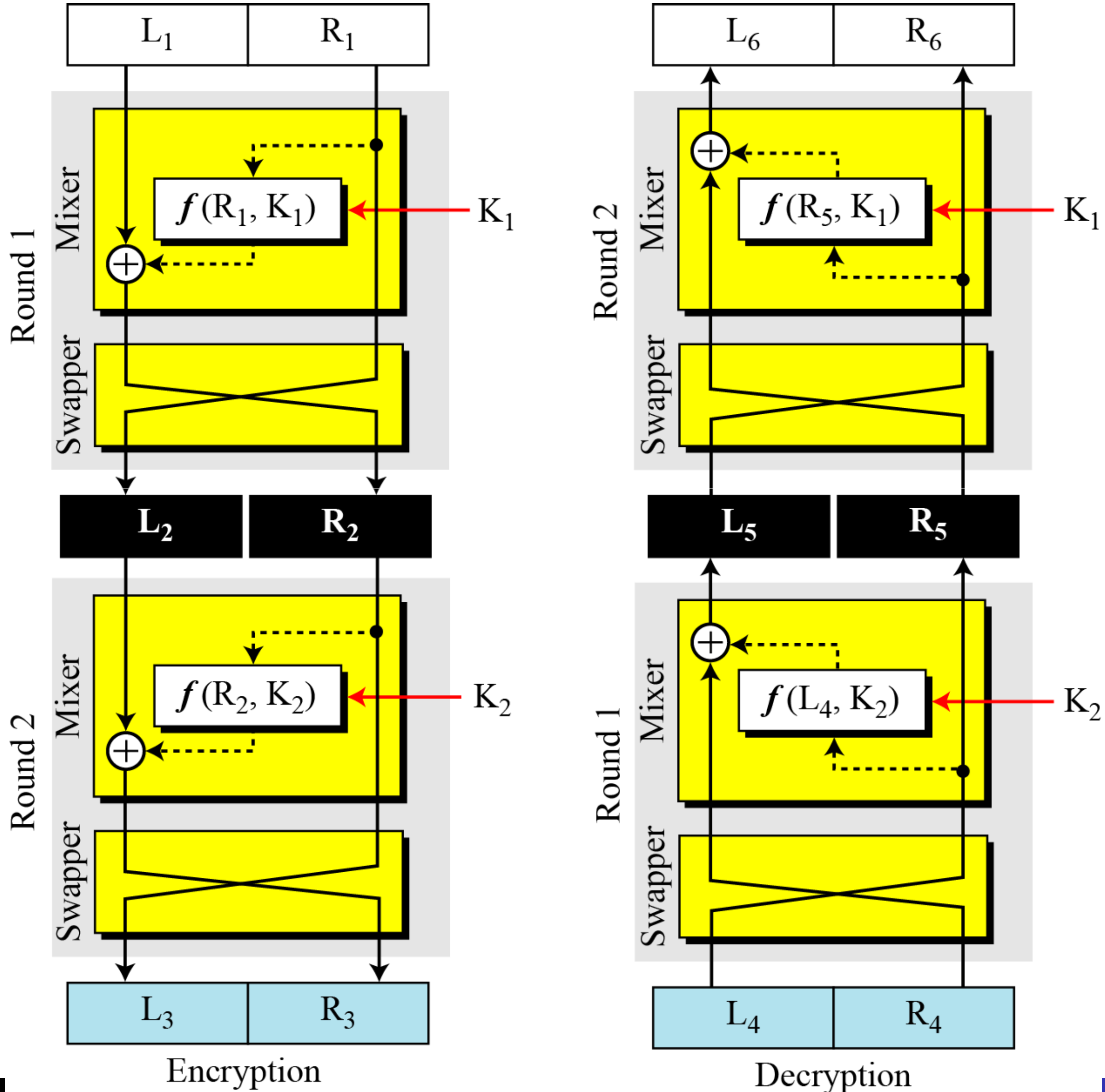
Decryption: $P = C \oplus f\,(K) = 1110 \oplus 1001 = 0111$

Two classes of product ciphers...

- Improvement in previous design



Final design of a fiestel cipher...



Final design of a fiestel cipher...

- Non-fiestel cipher
  - Used on invertible components
  - A component in the encryption cipher has the corresponding component in the decryption cipher.

Attacks on modern ciphers

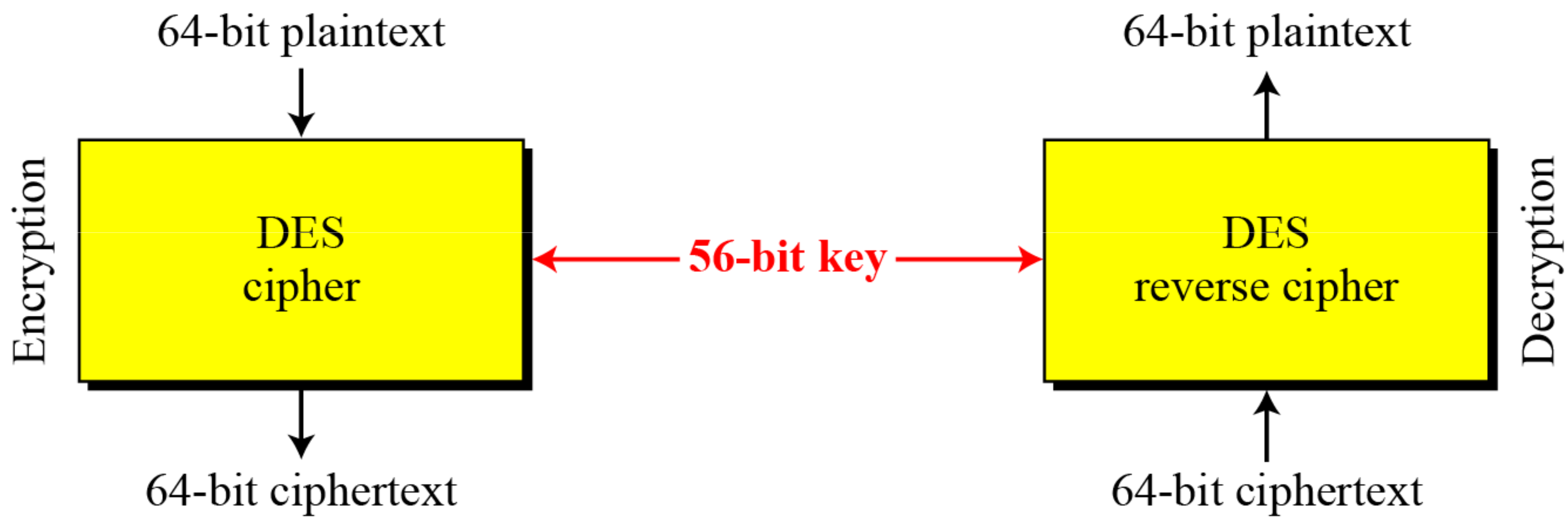
- Attacks on traditional ciphers can also be used on modern block ciphers, but today's block ciphers resist most of the attacks that are possible in classical ciphers

Data Encryption Standard (DES)

Introduction

- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).
- In 1973, NIST published a request for proposals for a national symmetric-key cryptosystem. A proposal from IBM, a modification of a project called Lucifer, was accepted as DES. DES was published in the Federal Register in March 1975 as a draft of the Federal Information Processing Standard (FIPS).

Overview

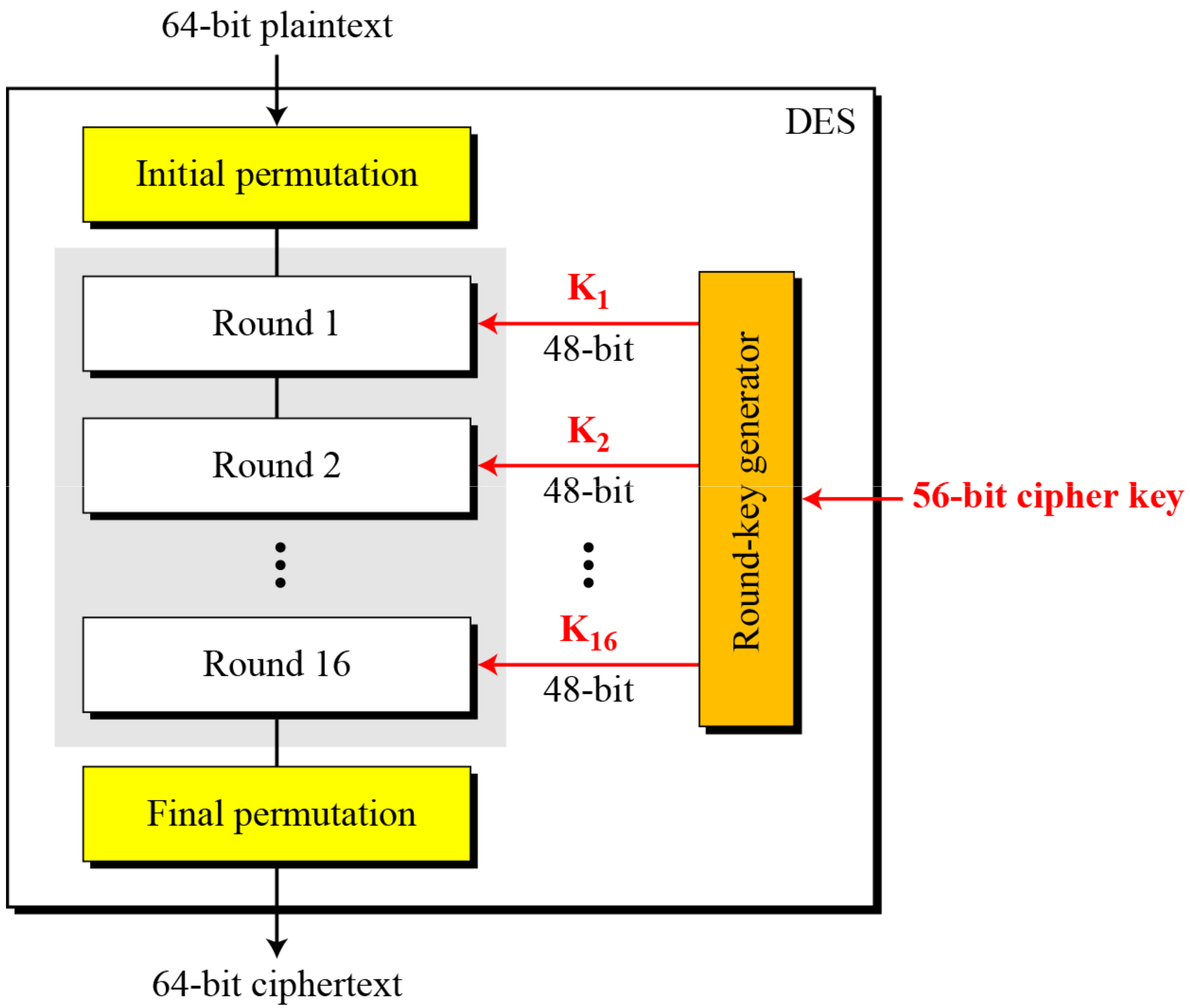


64-bit plaintext
Encryption
DES
cipher
56-bit key
64-bit ciphertext
64-bit plaintext
DES
reverse cipher
Decryption
64-bit ciphertext

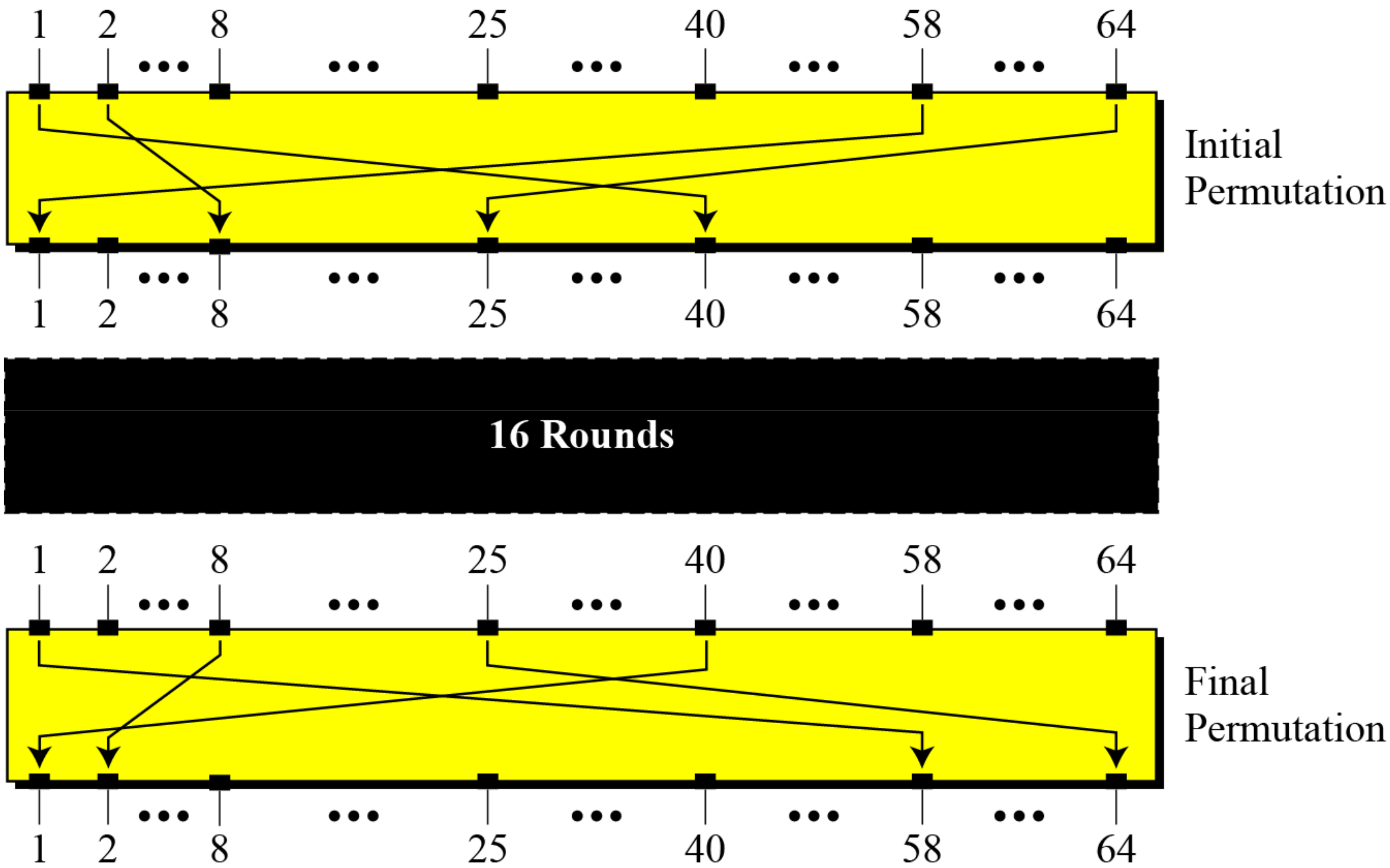
Structure of DES

- The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.

Structure of DES...



Initial and Final permutations



Initial and Final permutations...

Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

Initial and Final permutations...

- Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

Initial Permutation								*Final Permutation*							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25

Initial and Final permutations...

- Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

- Solution:

Assignment questions...

- Prove that the initial and final permutations are the inverse of each other by finding the output of the final permutation if the input is

0x0002 0000 0000 0001

Assignment questions...

- Prove that the initial and final permutations are the inverse of each other by finding the output of the final permutation if the input is

0x0002 0000 0000 0001

Assignment questions...

- The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

  C = P XOR f(K)   and P = C XOR f(K)

Assignment questions...

- The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

  C = P XOR f(K)   and P = C XOR f(K)

- Solution
  - The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

Encryption: $C = P \oplus f\ (K) = 0111 \oplus 1001 = 1110$

Decryption: $P = C \oplus f\ (K) = 1110 \oplus 1001 = 0111$

Assignment questions...

- A transposition block has 10 inputs and 10 outputs. What is the order of the permutation group? What is the key size?
- A substitution block has 10 inputs and 10 outputs. What is the order of the permutation group? What is the key size?
- The input/output relation in a 2x2 S-box is shown by the following table. Show the table for the inverse S-box.

Input left bit/input right bit	0	1
0	01	11
1	10	00

Assignment questions...

- Show the D-box defined by the following table:

8 1 2 3 4 5 6 7

- What is the pattern in the ciphertext of a one-time pad cipher in each of the following cases?

  a. The plaintext is made of n 0's.

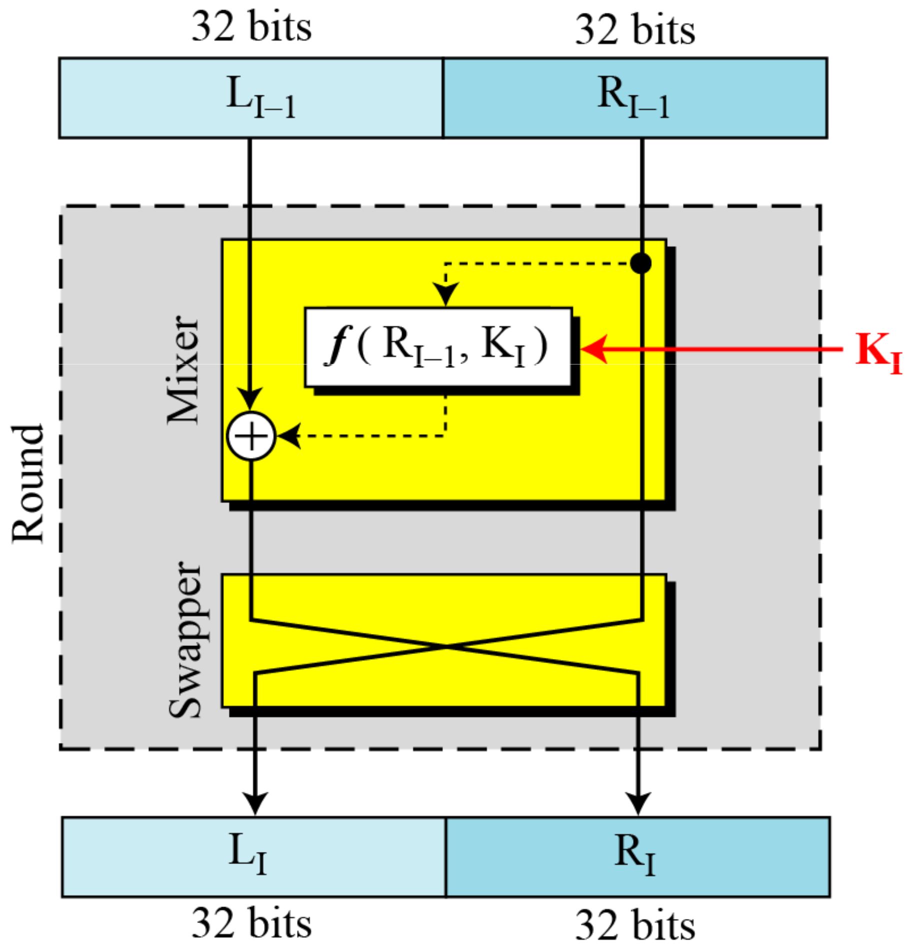
  b. The plaintext is made of n 1's.

  c. The plaintext is made of alternating 0's and 1's.

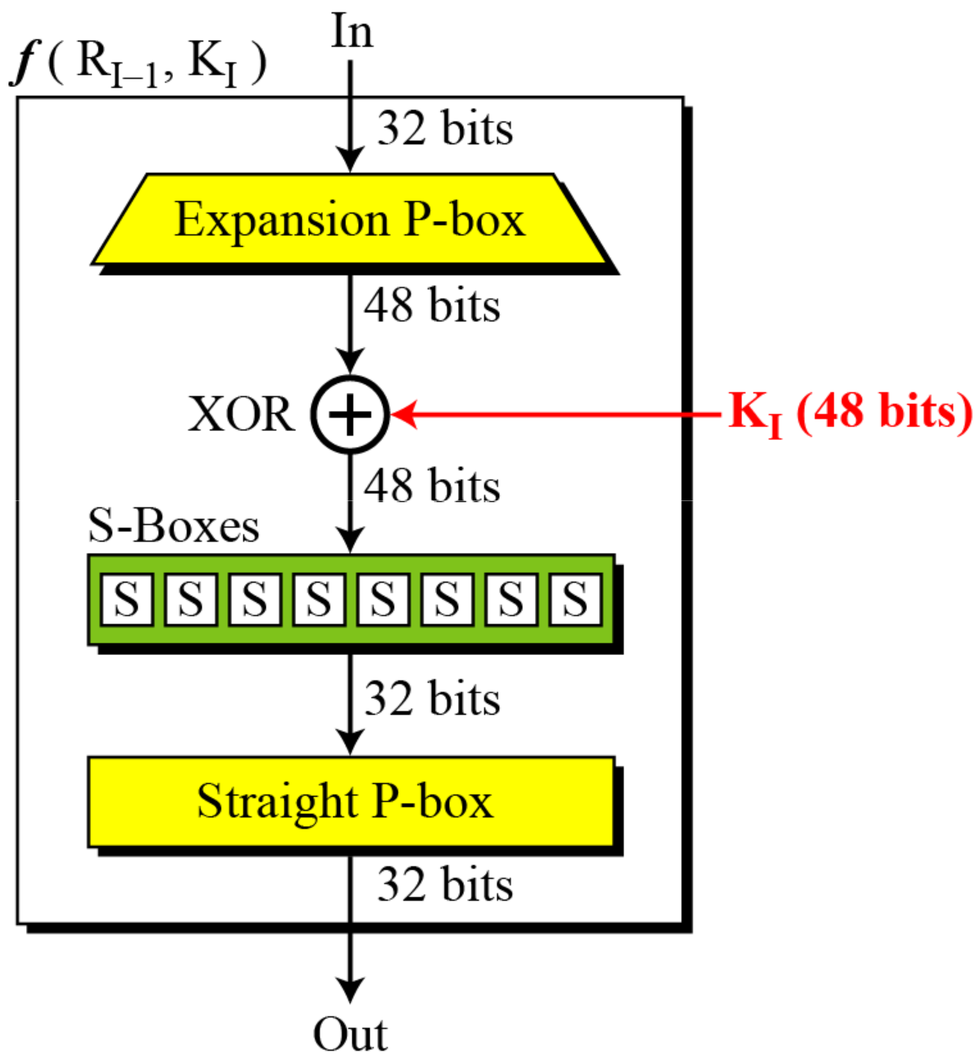
  d. The plaintext is a random string of bits.

Rounds

- 16 rounds of Fiestel cipher

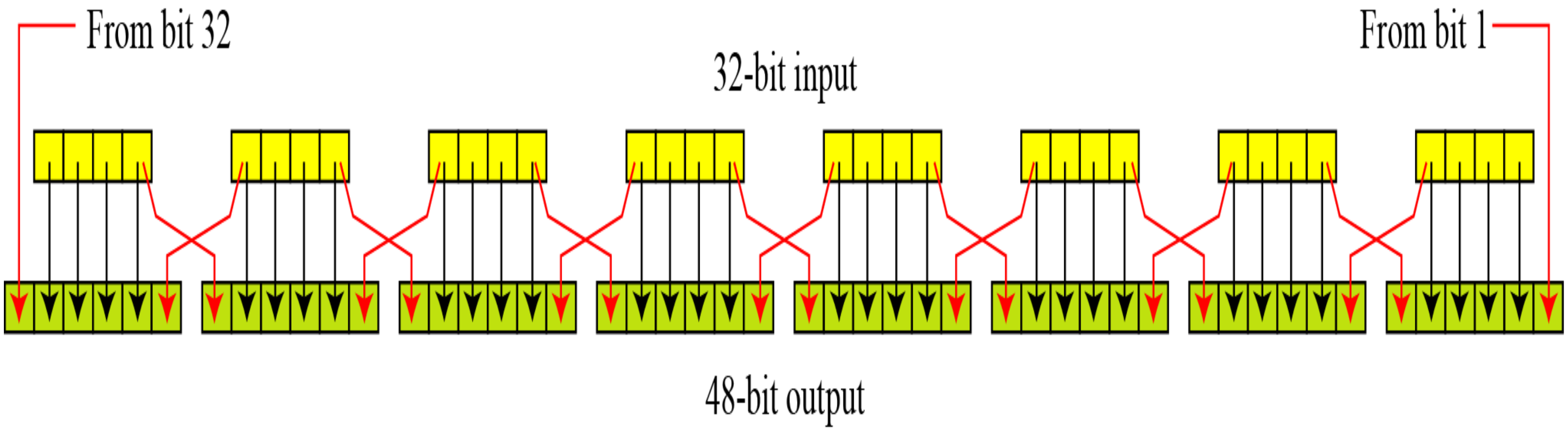


DES function



DES function…

- Expansion P-box
  - Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.



DES function...

- Expansion P-box
  - Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.

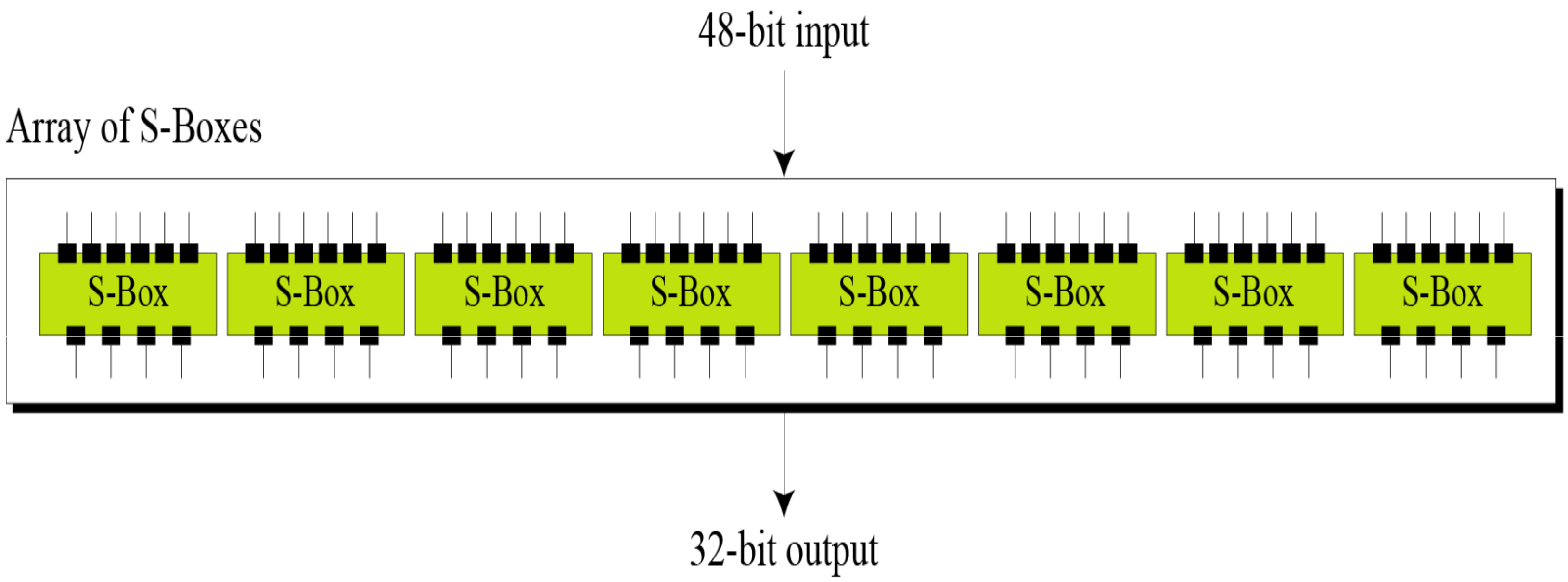
32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

DES function…

- Whitener
  - After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

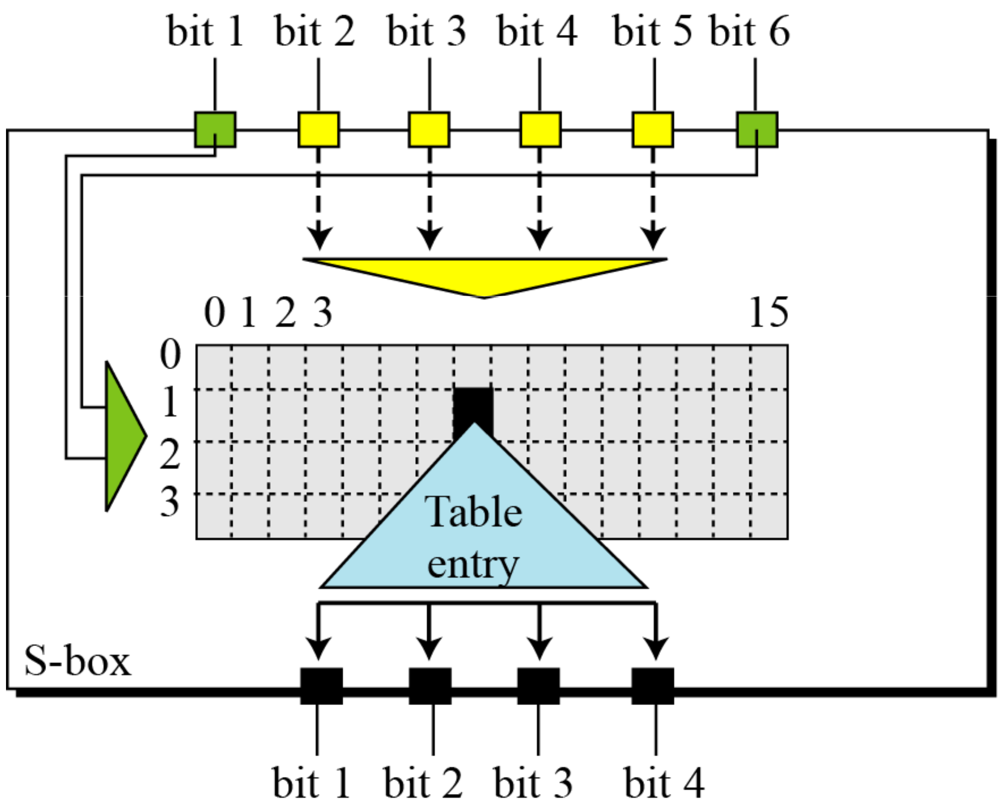
DES function…

- S-Boxes



DES function...

- S-Boxes



DES function…

- Permutations for S-Box-I

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

DES function…

- Example
  - The input to S-box 1 is 100011. What is the output?

DES function…

- Example
  - The input to S-box 1 is 100011. What is the output?
- Solution
  - If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table of S-box 1. The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

DES function...

- Straight P-Box

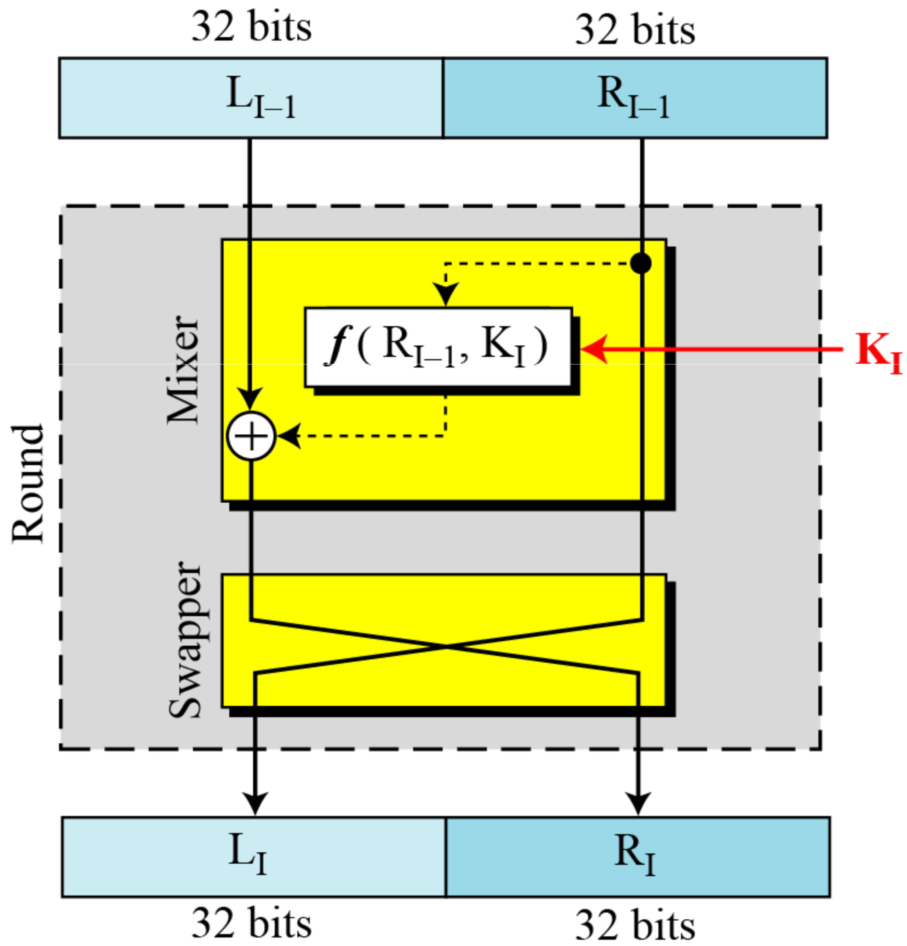
16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Cipher and reverse cipher

- Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.
- First Approach
  - To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

Cipher and reverse cipher...

- 16 rounds of Fiestel cipher



Cipher and reverse cipher...

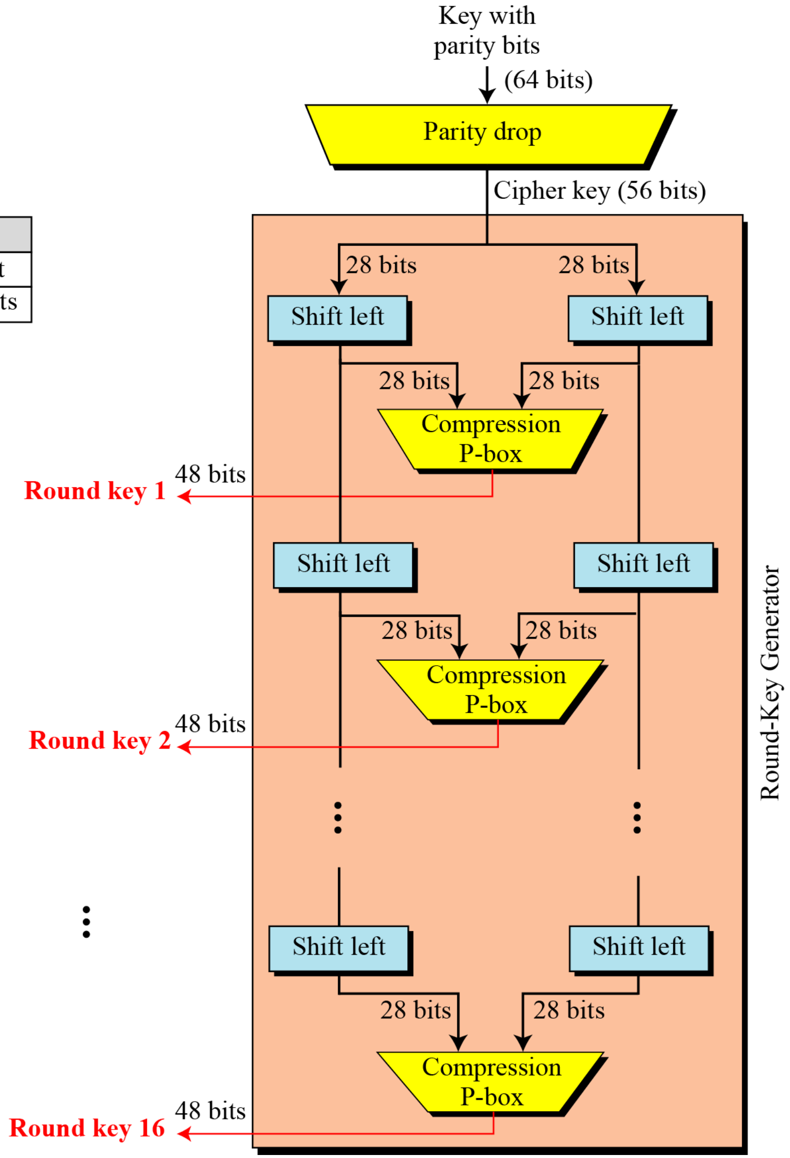
- Alternative approach
  - We can make all 16 rounds the same by including one swapper to the 16th round and add an extra swapper after that.

Key generation

- The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.

Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



Key generation

Parity bit drop table

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Number of shift bits

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Key generation

Key compression table

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

DES: an example of encipherment

Plaintext: 123456ABCD132536 Key: AABB09182736CCDD

CipherText: C0B7A8D05F3A829C

Plaintext: 123456ABCD132536			
After initial permutation: 14A7D67818CA18AD After splitting: L_0=14A7D678 R_0=18CA18AD			
Round	*Left*	*Right*	*Round Key*
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

DES: an example of encipherment...

Round 5	236779C2	A15A4B87	69A629FEC913
Round 6	A15A4B87	2E8F9C65	C1948E87475E
Round 7	2E8F9C65	A9FC20A3	708AD2DDB3C0
Round 8	A9FC20A3	308BEE97	34F822F0C66D
Round 9	308BEE97	10AF9D37	84BB4473DCCC
Round 10	10AF9D37	6CA6CB20	02765708B5BF
Round 11	6CA6CB20	FF3C485F	6D5560AF7CA5
Round 12	FF3C485F	22A5963B	C2C1E96A4BF3
Round 13	22A5963B	387CCDAA	99C31397C91F
Round 14	387CCDAA	BD2DD2AB	251B8BC717D0
Round 15	BD2DD2AB	CF26B472	3330C5D9A36D
Round 16	19BA9212	CF26B472	181C5D75C66D

After combination: 19BA9212CF26B472

Ciphertext: C0B7A8D05F3A829C *(after final permutation)*

DES: an example of decipherment

Ciphertext: C0B7A8D05F3A829C			
After initial permutation: 19BA9212CF26B472 After splitting: L_0=19BA9212 R_0=CF26B472			
Round	*Left*	*Right*	*Round Key*
Round 1	CF26B472	BD2DD2AB	181C5D75C66D
Round 2	BD2DD2AB	387CCDAA	3330C5D9A36D
. . .	. . .	. . .	. . .
Round 15	5A78E394	18CA18AD	4568581ABCCE
Round 16	14A7D678	18CA18AD	194CD072DE8C
After combination: 14A7D67818CA18AD			
Plaintext:123456ABCD132536 (after final permutation)			

DES: Analysis

- Two desirable properties
  - Avalanche effect
  - Completeness

Avalanche effect

- To check the avalanche effect in DES, let us encrypt two plaintext blocks (with the same key) that differ only in one bit and observe the differences in the number of bits in each round.

Plaintext: 0000000000000000 Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

Plaintext: 000000000000000<u>**1**</u> Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3

Avalanche effect...

- Ciphertext blocks differ in 29 bits.
  - i.e. changing approximately 1.5 percent of the plaintext creates a change of approximately 45 percent in the ciphertext.

Rounds	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
Bit differences	1	6	20	29	30	33	32	29	32	39	33	28	30	31	30	29

Completeness

- Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext.
  - S-Box
  - P-Box
  - 16 rounds of fiestel blocks

DES Weaknesses

- Weakness in key

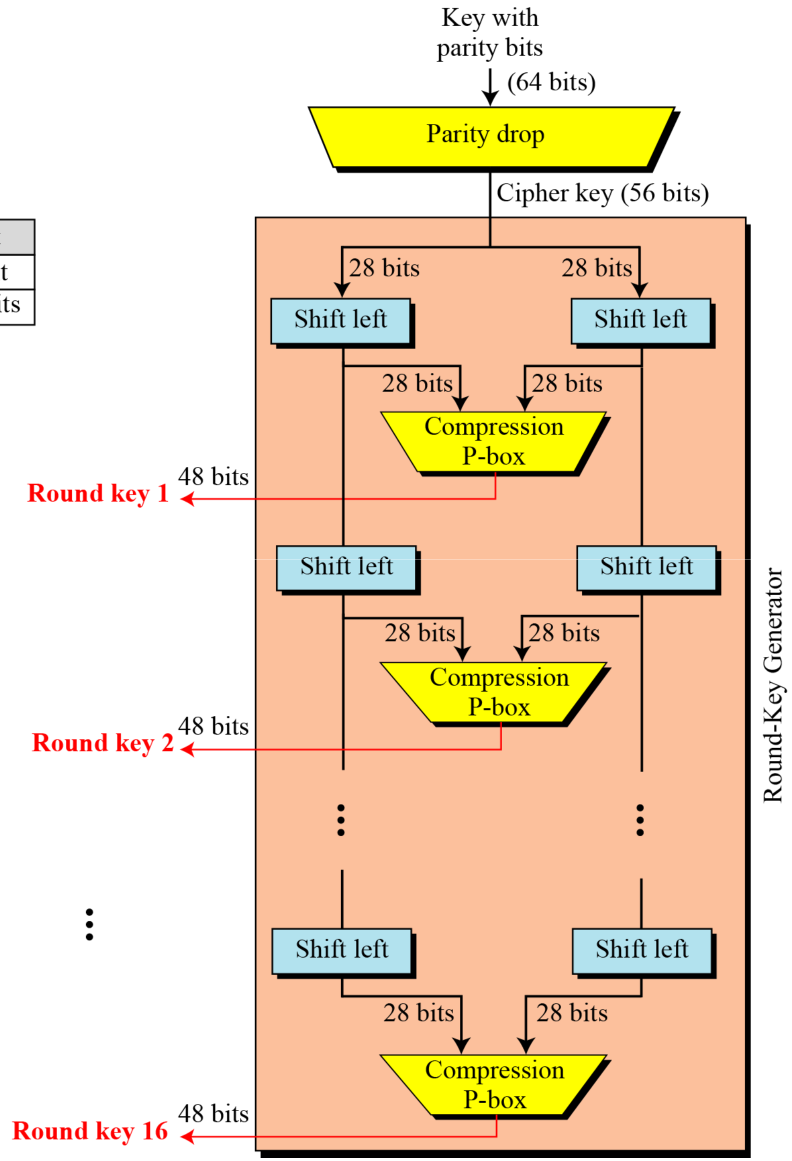
Table 6.18 *Weak keys*

Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

Key generation

Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



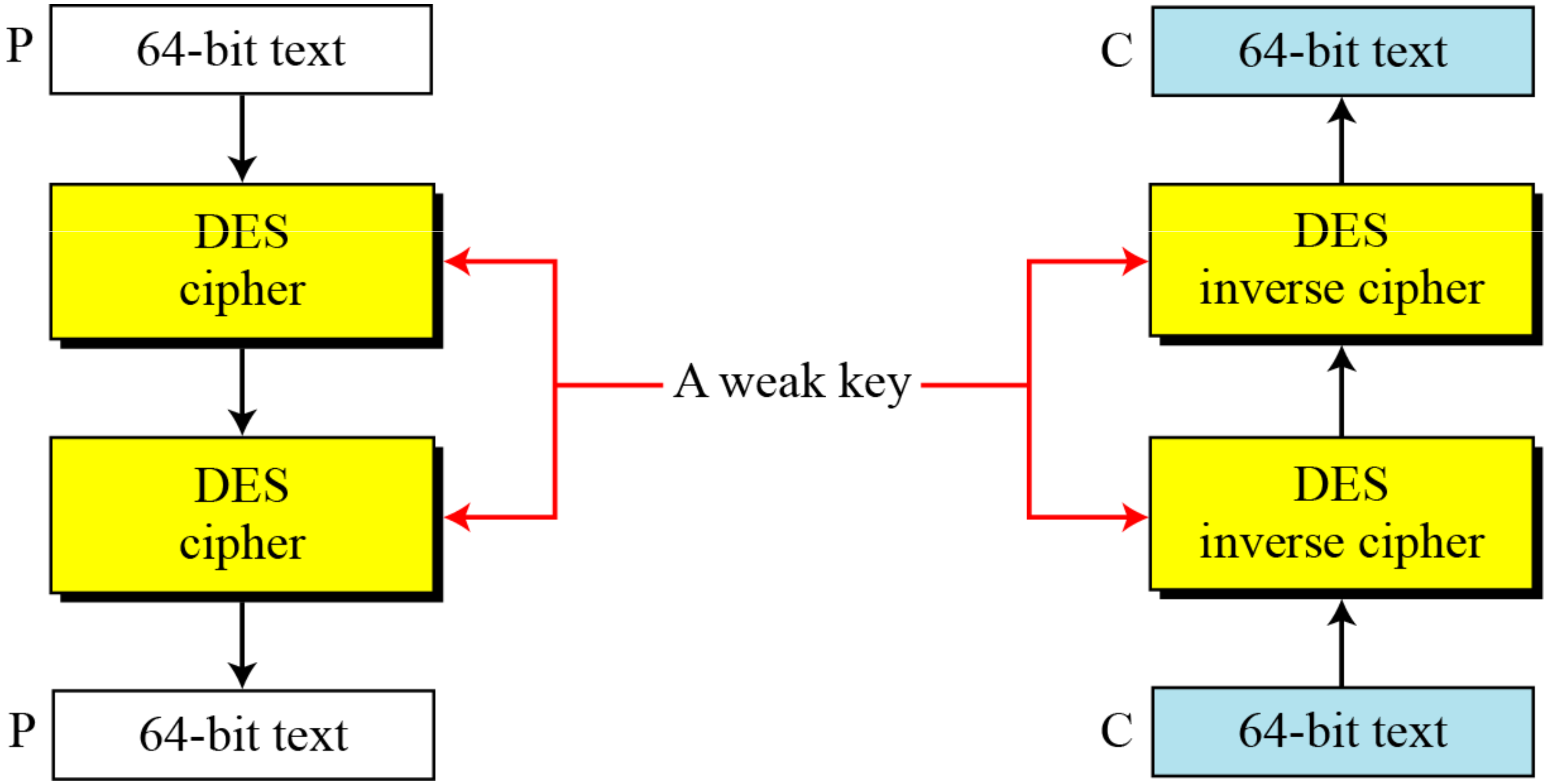
DES Weaknesses...

- Example
  - After two encryptions with the same key the original plaintext block is created. Note that we have used the encryption algorithm two times, not one encryption followed by another decryption.

Key: 0x0101010101010101	
Plaintext: *0x1234567887654321*	Ciphertext: 0x814FE938589154F7
Key: 0x0101010101010101	
Plaintext: 0x814FE938589154F7	Ciphertext: *0x1234567887654321*

DES Weaknesses...

- Double encryption and decryption with a weak key



DES Weaknesses...

- Semi weak keys

Table 6.19 *Semi-weak keys*

First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

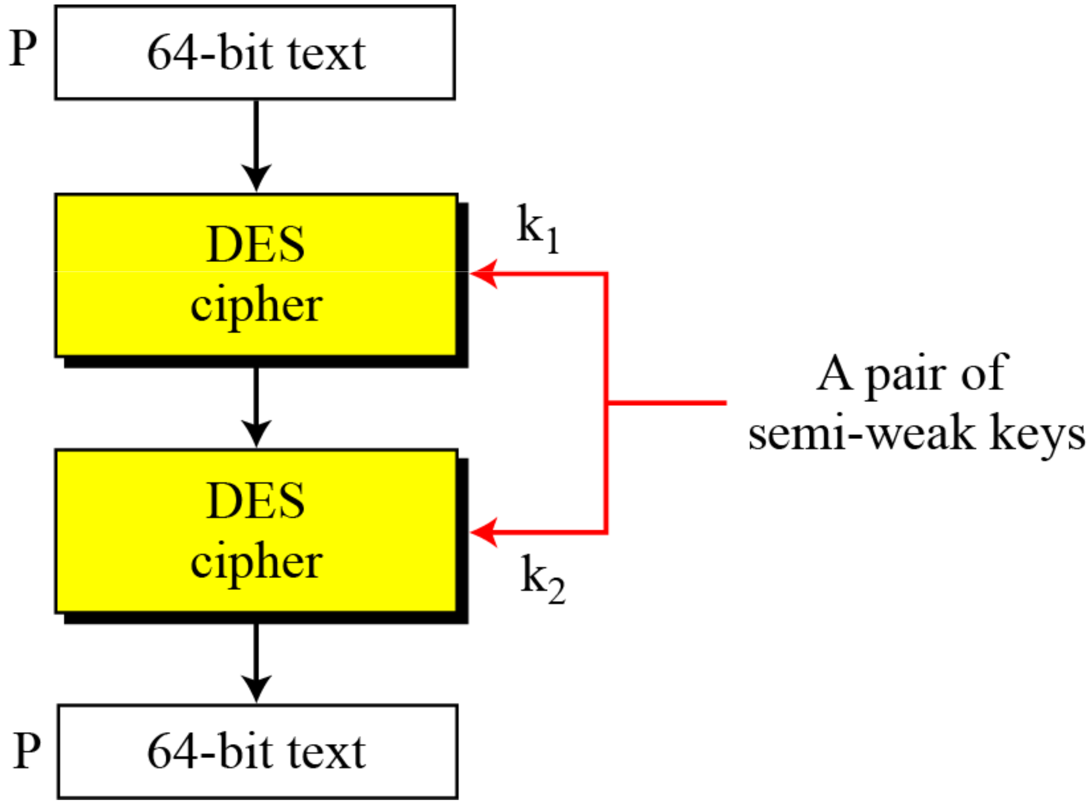
DES Weaknesses...

- Semi weak keys...

Round key 1	9153E54319BD	6EAC1ABCE642
Round key 2	6EAC1ABCE642	9153E54319BD
Round key 3	6EAC1ABCE642	9153E54319BD
Round key 4	6EAC1ABCE642	9153E54319BD
Round key 5	6EAC1ABCE642	9153E54319BD
Round key 6	6EAC1ABCE642	9153E54319BD
Round key 7	6EAC1ABCE642	9153E54319BD
Round key 8	6EAC1ABCE642	9153E54319BD
Round key 9	9153E54319BD	6EAC1ABCE642
Round key 10	9153E54319BD	6EAC1ABCE642
Round key 11	9153E54319BD	6EAC1ABCE642
Round key 12	9153E54319BD	6EAC1ABCE642
Round key 13	9153E54319BD	6EAC1ABCE642
Round key 14	9153E54319BD	6EAC1ABCE642
Round key 15	9153E54319BD	6EAC1ABCE642
Round key 16	6EAC1ABCE642	9153E54319BD

DES Weaknesses...

- A pair of semi-weak keys in encryption and decryption



DES Weaknesses...

- What is the probability of randomly selecting a weak, a semi-weak, or a possible weak key?
  - DES has a key domain of 256. The total number of the above keys are 64 (4 + 12 + 48). The probability of choosing one of these keys is 8.8×10^{-16}, almost impossible.

DES Weaknesses...

- Key complement

Key Complement In the key domain (2^{56}), definitely half of the keys are *complement* of the other half. A **key complement** can be made by inverting (changing 0 to 1 or 1 to 0) each bit in the key. Does a key complement simplify the job of the cryptanalysis? It happens that it does. Eve can use only half of the possible keys (2^{55}) to perform brute-force attack. This is because

$$C = E(K, P) \rightarrow \overline{C} = E(\overline{K}, \overline{P})$$

In other words, if we encrypt the complement of plaintext with the complement of the key, we get the complement of the ciphertext. Eve does not have to test all 2^{56} possible keys, she can test only half of them and then complement the result.

DES Weaknesses...

- Key complement example

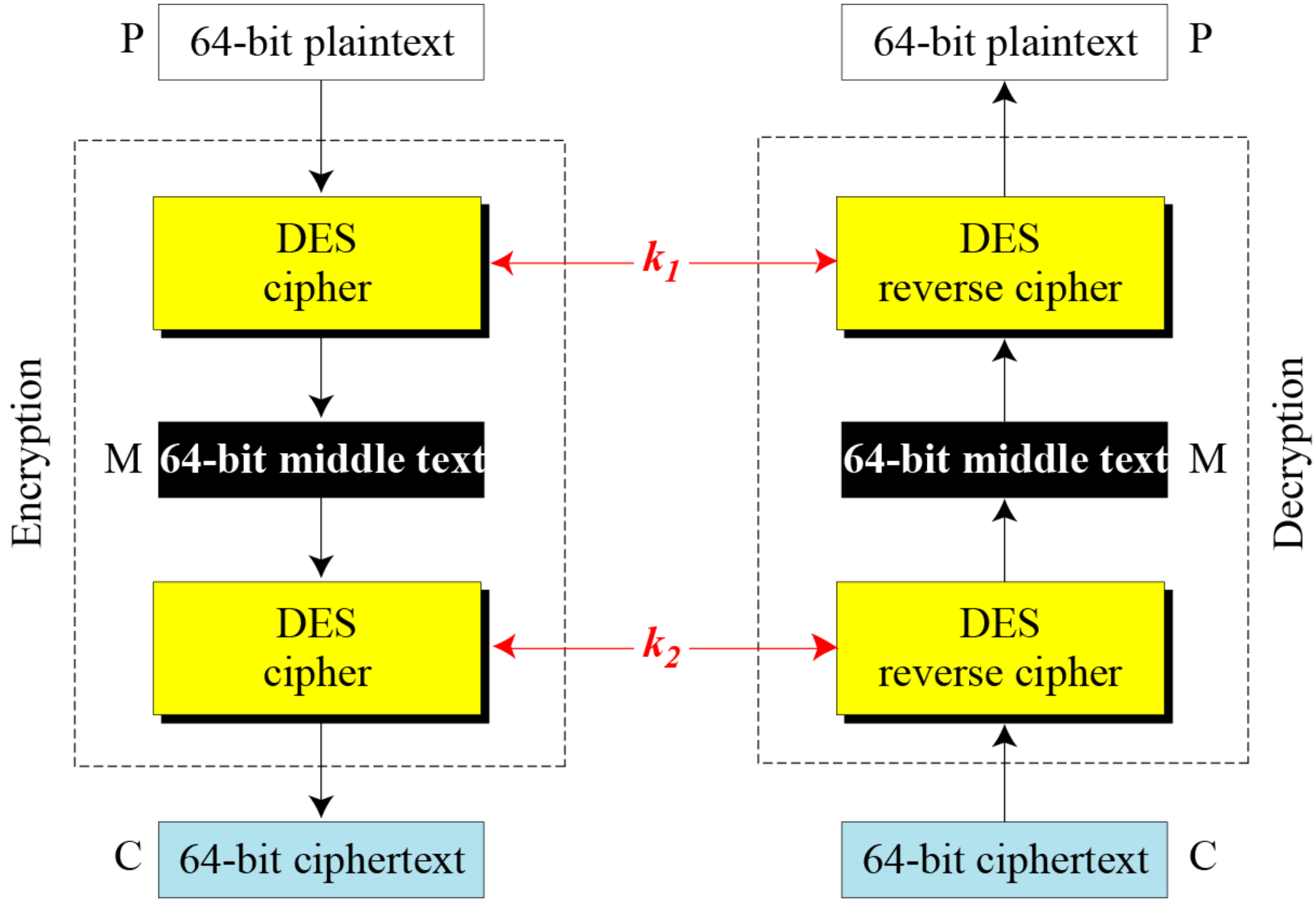
Table 6.20 *Results for Example 6.10*

	Original	*Complement*
Key	1234123412341234	EDCBEDCBEDCBEDCB
Plaintext	12345678ABCDEF12	EDCBA987543210ED
Ciphertext	E112BE1DEFC7A367	1EED41E210385C98

Multiple DES

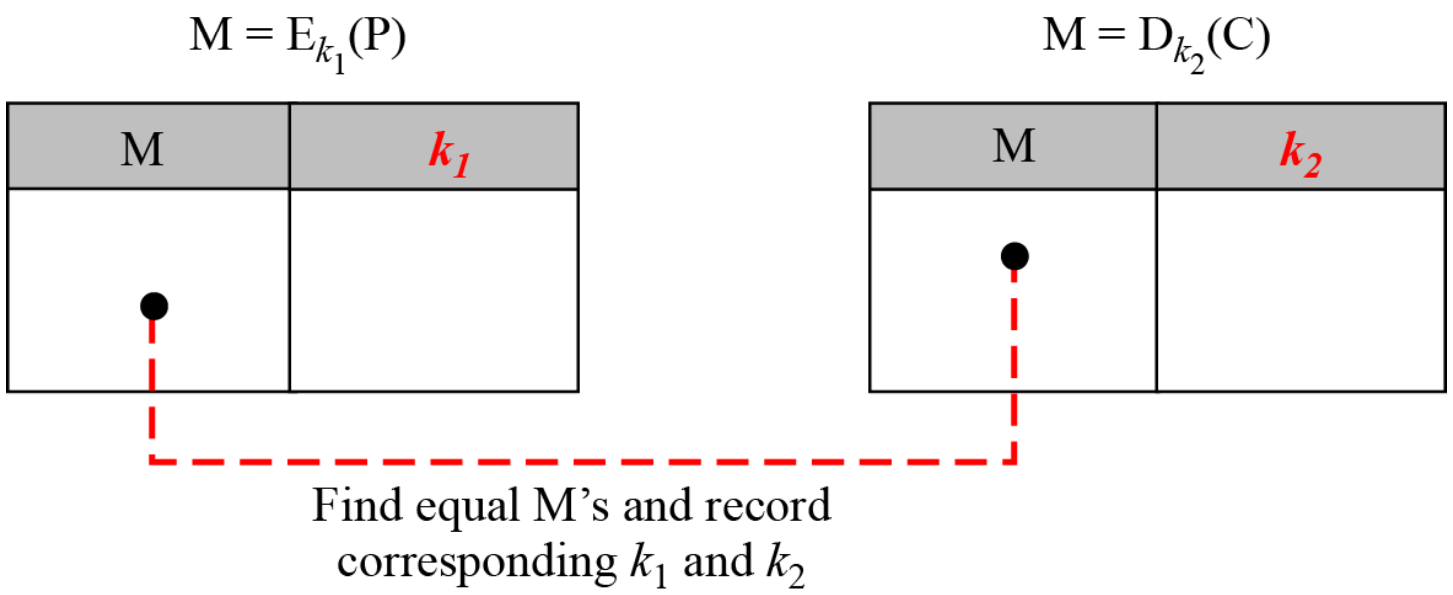
- The major criticism of DES
  - its key length.
- Fortunately DES is not a group.
  - This means that we can use double or triple DES to increase the key size.

Double DES

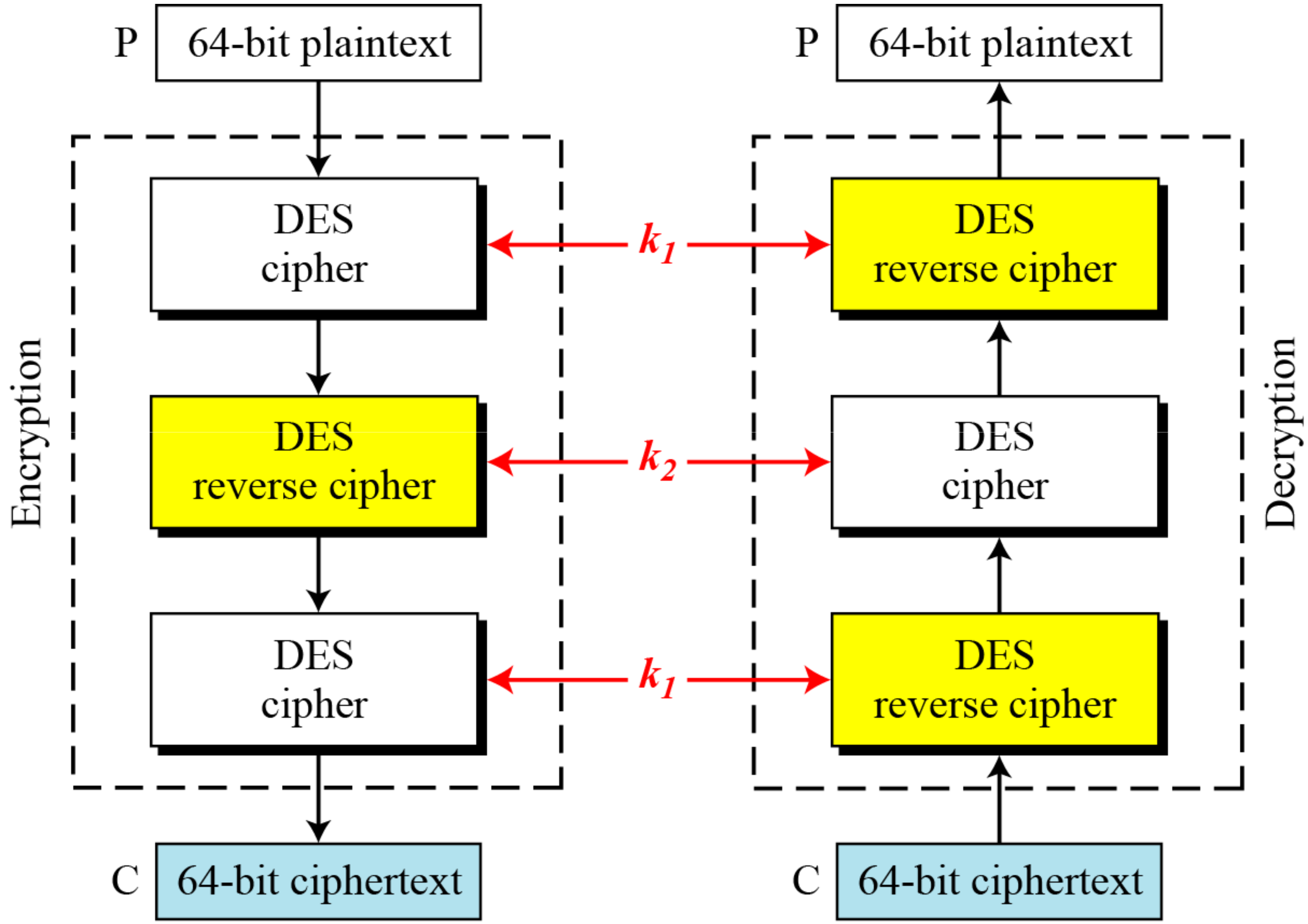

P
64-bit plaintext
DES
cipher
Encryption
M
64-bit middle text
DES
cipher
C
64-bit ciphertext
k1
k2
64-bit plaintext
P
DES
reverse cipher
64-bit middle text
M
Decryption
DES
reverse cipher
64-bit ciphertext
C

Double DES

- Meet-in-the-Middle Attack
  - a known-plaintext attack
  - double DES improves this vulnerability slightly (to 2^{57} tests), but not tremendously (to 2^{112}).



Triple DES


P
64-bit plaintext
DES
cipher
DES
reverse cipher
DES
cipher
C
64-bit ciphertext
Encryption
k_1
k_2
k_1
P
64-bit plaintext
DES
reverse cipher
DES
cipher
DES
reverse cipher
C
64-bit ciphertext
Decryption